



# **Service Guide for Mobile Broadcast Services**

Approved Version 1.2 – 31 Jan 2017

---

**Open Mobile Alliance**  
OMA-TS-BCAST\_Service\_Guide-V1\_2-20170131-A

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

**NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.**

**THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.**

© 2017 Open Mobile Alliance All Rights Reserved.

Used with the permission of the Open Mobile Alliance Under the terms set forth above.

# Contents

<b>1. SCOPE</b> .....	<b>9</b>
<b>2. REFERENCES</b> .....	<b>10</b>
<b>2.1 NORMATIVE REFERENCES</b> .....	<b>10</b>
<b>2.2 INFORMATIVE REFERENCES</b> .....	<b>13</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>14</b>
<b>3.1 CONVENTIONS</b> .....	<b>14</b>
<b>3.2 DEFINITIONS</b> .....	<b>14</b>
<b>3.3 ABBREVIATIONS</b> .....	<b>15</b>
<b>4. INTRODUCTION</b> .....	<b>17</b>
<b>4.1 VERSION 1.0</b> .....	<b>17</b>
<b>4.2 VERSION 1.1</b> .....	<b>17</b>
<b>4.3 VERSION 1.2</b> .....	<b>17</b>
<b>5. SERVICE GUIDE</b> .....	<b>18</b>
<b>5.1 SERVICE GUIDE DATA MODEL</b> .....	<b>18</b>
5.1.1 Service Guide Structure and Fragmentation .....	18
5.1.2 Service Guide Elements, Attributes and Meta Data.....	22
5.1.2.1 <i>Service</i> .....	23
5.1.2.2 <i>Schedule</i> .....	44
5.1.2.3 <i>Content</i> .....	52
5.1.2.4 <i>Access</i> .....	75
5.1.2.5 <i>Session Description Information</i> .....	101
5.1.2.6 <i>Purchase Item</i> .....	107
5.1.2.7 <i>Purchase Data</i> .....	114
5.1.2.8 <i>Purchase Channel</i> .....	131
5.1.2.9 <i>PreviewData</i> .....	137
5.1.2.10 <i>InteractivityData</i> .....	147
<b>5.2 ANNOUNCING SERVICE GUIDES WITHIN A SERVICE GUIDE</b> .....	<b>156</b>
<b>5.3 INTERFACES FOR SERVICE GUIDE GENERATION</b> .....	<b>157</b>
5.3.1 Service Guide Data Model as Generic Interchange format.....	157
<b>5.4 SERVICE GUIDE DELIVERY</b> .....	<b>157</b>
5.4.1 Encapsulating and identifying fragments.....	158
5.4.1.1 <i>Fragment identifier versus transport identifier</i> .....	158
5.4.1.2 <i>Resolving fragment references</i> .....	159
5.4.1.3 <i>Service Guide fragment encapsulation</i> .....	159
5.4.1.4 <i>Compression of Service Guide Delivery Units and Service Guide Delivery Descriptors</i> .....	162
5.4.1.5 <i>Service Guide Delivery Descriptor</i> .....	163
5.4.2 Delivery over the Broadcast Channel .....	200
5.4.2.1 <i>Signaling Changes in the Service Guide over Broadcast Channel</i> .....	201
5.4.3 Delivery over the Interaction Channel .....	202
5.4.3.1 <i>Rules applicable to all requests and responses</i> .....	203
5.4.3.2 <i>Unspecific request for retrieving service guide over Interaction Channel</i> .....	207
5.4.3.3 <i>Requests for specific Service Guide Delivery Descriptors</i> .....	207
5.4.3.4 <i>Requests for specific Service Guide fragments</i> .....	209
5.4.3.5 <i>Retrieving files referenced within Service Guide</i> .....	216
<b>5.5 SERVICE GUIDE UPDATE AND MANAGEMENT</b> .....	<b>217</b>
5.5.1 Service Guide Update and Management using the Broadcast Channel .....	217
5.5.1.1 <i>Localized changes and their discovery (Informative)</i> .....	218
5.5.1.2 <i>Enabling Terminal to determine Service Guide completeness</i> .....	221
5.5.1.3 <i>Terminal behaviour determining the Service Guide completeness (Informative)</i> .....	222
5.5.2 Service Guide update and management using the Interaction Channel .....	223
5.5.3 Service Guide Update and Management Cases for Hybrid Broadcast/Interactive Scenario (Informative).....	224
5.5.3.1 <i>Service Guide partly Distributed over Broadcast Channel and partly Distributed over Interaction Channel</i> .....	224
5.5.3.2 <i>Service Guide completely Distributed over both Broadcast Channel and Interaction Channel</i> .....	224
5.5.3.3 <i>Service Guide Distributed completely over Interaction Channel</i> .....	224
<b>5.6 SERVICE GUIDE BACKEND INTERFACES</b> .....	<b>225</b>

5.6.1	Service Guide management in the backend .....	225
5.6.2	Service Guide contents exchange.....	225
5.6.2.1	Protocol Stacks .....	225
5.6.2.2	Service Guide content delivery.....	226
5.6.2.3	Service Protection Information Requests.....	229
5.6.2.4	Service Provisioning Information Request.....	231
<b>5.7</b>	<b>USE OF PREVIEW DATA TO SUPPORT SERVICE SWITCHING, BROWSING AND PREVIEW .....</b>	<b>232</b>
5.7.1	Preview Data for Service-by-Service Switching.....	234
5.7.2	Preview Data for Service Guide Browsing .....	235
5.7.3	Preview Data for Service Preview .....	237
5.7.4	Preview Data for Barker .....	238
5.7.5	Preview Data as an alternative to blackout .....	239
<b>5.8</b>	<b>ASSOCIATING ACCESSES WITH SERVICES AND CONTENT.....</b>	<b>240</b>
5.8.1	Multiple accesses .....	240
5.8.1.1	Choices of access .....	240
5.8.2	Association between 'Service' and 'Access' fragment.....	241
5.8.3	Association between 'Service' and 'Access' fragment through 'Schedule' fragment.....	241
5.8.4	Association between 'Content' and 'Access' fragment through 'Service' or 'Schedule' fragment.....	242
5.8.4.1	Content access through service access .....	242
5.8.4.2	Content specific access via 'Schedule' fragment .....	243
5.8.5	Use of 'defaultSchedule' and 'onDemand' attributes .....	244
5.8.6	Terminal interpretation for content consumption.....	244
5.8.7	Distribution Window and Presentation Window for Cachecast Services .....	246
<b>6.</b>	<b>DISCOVERY OF SERVICE GUIDE .....</b>	<b>247</b>
<b>6.1</b>	<b>SERVICE GUIDE TRANSMITTED OVER BROADCAST CHANNEL .....</b>	<b>247</b>
6.1.1	Service Guide Discovery over Broadcast Channel .....	247
6.1.2	Service Guide Discovery over Interaction Channel .....	247
<b>6.2</b>	<b>SERVICE GUIDE TRANSMITTED OVER INTERACTION CHANNEL.....</b>	<b>248</b>
6.2.1	Service Guide Discovery using DNS.....	248
6.2.1.1	Derivation of Domain Name from IMSI.....	249
<b>7.</b>	<b>MULTI-LANGUAGE SUPPORT .....</b>	<b>250</b>
<b>7.1</b>	<b>DECLARING SERVICE GUIDE ELEMENTS IN DIFFERENT LANGUAGES IN THE SERVICE GUIDE .....</b>	<b>250</b>
<b>7.2</b>	<b>DECLARING COMPONENTS OF THE BROADCAST SERVICE IN DIFFERENT LANGUAGES IN THE SERVICE GUIDE .....</b>	<b>251</b>
7.2.1	Service and content specialization with audio tracks and sub-titles.....	251
<b>APPENDIX A.</b>	<b>CHANGE HISTORY (INFORMATIVE).....</b>	<b>253</b>
<b>A.1</b>	<b>APPROVED VERSION HISTORY .....</b>	<b>253</b>
<b>APPENDIX B.</b>	<b>STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....</b>	<b>254</b>
<b>B.1</b>	<b>SCR FOR BCAST SERVICE GUIDE CLIENT (SG-C) .....</b>	<b>254</b>
<b>B.2</b>	<b>SCR FOR BCAST SERVICE GUIDE GENERATION/ADAPTATION/DISTRIBUTION (SG-GAD).....</b>	<b>255</b>
<b>B.3</b>	<b>SCR FOR BCAST SERVICE GUIDE SUBSCRIPTION SOURCE (SGSS).....</b>	<b>257</b>
<b>B.4</b>	<b>SCR FOR BCAST SERVICE GUIDE APPLICATION SOURCE (SGAS).....</b>	<b>257</b>
<b>APPENDIX C.</b>	<b>SERVICE GUIDE APPLICATION SCENARIOS (INFORMATIVE) .....</b>	<b>259</b>
<b>C.1</b>	<b>SCENARIO OF SELECTIVE VIEWING BASED ON THE CODEC REQUIREMENT IN THE SG.....</b>	<b>259</b>
<b>C.2</b>	<b>SCENARIO OF SELECTIVE VIEWING BASED ON THE BANDWIDTH REQUIREMENT IN THE SG.....</b>	<b>259</b>
<b>C.3</b>	<b>SCENARIO OF A TV PROGRAM WHICH IS BROADCASTED IN TWO LANGUAGES.....</b>	<b>259</b>
<b>C.4</b>	<b>SCENARIO OF A BROADCAST STREAMING SERVICE PROTECTED BY MULTIPLE KEY MANAGEMENT SYSTEMS .....</b>	<b>261</b>
<b>C.5</b>	<b>SCENARIO OF A MAIN SERVICE WITH INTERACTION.....</b>	<b>263</b>
C.5.1	Service (//this.example.com/service/401) .....	264
C.5.2	Schedule (//this.example.com/schedule/501).....	264
C.5.3	Content (//this.example.com/content/601) .....	265
C.5.4	InteractivityData (//this.example.com/int/801) .....	265
C.5.5	Access (//this.example.com/access/901).....	266
C.5.6	Access (//this.example.com/access/902).....	266
<b>C.6</b>	<b>SCENARIO OF A MAIN SERVICE WITH PURCHASABLE PROTECTED AUXILIARY DATA CONTENT.....</b>	<b>266</b>
<b>C.7</b>	<b>SCENARIO OF MULTIPLE SERVICES PROVIDED WITH SINGLE CHANNEL EACH .....</b>	<b>268</b>

**C.8 SCENARIO OF BUNDLING MULTIPLE CHANNELS INTO A SINGLE SERVICE .....269**

**C.9 SCENARIO OF USING SINGLE SERVICE GUIDE TO PROVIDE SERVICE DESCRIPTIONS FOR MULTIPLE SERVICE PROVIDERS.....271**

**C.10 SCENARIO OF SERVICE GUIDE ANNOUNCING OTHER SERVICE GUIDES .....274**

**C.11 SCENARIO OF FLUTE FILE DOWNLOAD SERVICE WITH FILE DESCRIPTIONS IN THE SERVICE GUIDE .....275**

**C.12 SCENARIOS OF UNICAST SERVICE GUIDE REQUESTS.....277**

    C.12.1 LIVE TV services .....277

        C.12.1.1 Service plan and provisioning information retrieval ..... 277

        C.12.1.2 Electronic Program Guide information ..... 278

        C.12.1.3 Preview information retrieval.....279

    C.12.2 ON DEMAND SERVICES .....280

        C.12.2.1 Audio Video On Demand services retrieval.....280

        C.12.2.2 On demand contents retrieval.....280

        C.12.2.3 Preview information retrieval.....281

    C.12.3 CACHE CAST DOWNLOADING .....281

        C.12.3.1 Service plan and provisioning information retrieval ..... 281

        C.12.3.2 Cachecast content retrieval.....282

        C.12.3.3 Preview information retrieval.....283

    C.12.4 General practice applicable to all scenarios .....283

**APPENDIX D. EXAMPLE OF SGDD AND SGDU .....284**

**APPENDIX E. SERVICE CLASS IDENTIFIER HANDLING (NORMATIVE).....285**

**E.1 DEFINITION OF SERVICE CLASS VALUES.....285**

        E.1.1 Service Class Definition .....285

        E.1.2 Service Class Naming and Registration .....285

**E.2 SERVICE CLASSES DEFINED IN OMA BCAST .....285**

        E.2.1 urn:oma:bcast:oma\_bsc:sg:1.0.....285

        E.2.2 urn:oma:bcast:oma\_bsc:fc:1.0 .....285

        E.2.3 urn:oma:bcast:oma\_bsc:tp:1.0 .....286

        E.2.4 urn:oma:bcast:oma\_bsc:nt:1.0 .....286

        E.2.5 urn:oma:bcast:oma\_bsc:csg:1.0 .....286

        E.2.6 urn:oma:bcast:oma\_bsc:st:1.0.....286

        E.2.7 urn:oma:bcast:oma\_bsc:rifc:1.0 .....286

        E.2.8 urn:oma:bcast:oma\_bsc:fd:1.1 .....286

        E.2.9 urn:oma:bcast:oma\_bsc:sp:1.1 .....286

**APPENDIX F. GLOBAL STATUS CODES .....287**

**APPENDIX G. MIME MEDIA TYPES .....288**

**G.1 MEDIA-TYPE REGISTRATION REQUEST FOR APPLICATION/VND.OMA.BCAST.SGDU .....288**

**G.2 MEDIA-TYPE REGISTRATION REQUEST FOR APPLICATION/VND.OMA.BCAST.SGDD+XML .....289**

**APPENDIX H. IDENTIFICATION OF UNIQUE ELEMENTS (NORMATIVE) .....291**

**H.1 DEFINITION OF BCAST IDENTIFICATION AUTHORITY .....291**

**H.2 IDENTIFICATION OF CONTENT WITH BASECID .....291**

**APPENDIX I. SERVICE GUIDE DELIVERY IN HYBRID BROADCAST/INTERACTIVE SCENARIOS (INFORMATIVE)293**

**I.1 FILE AND STREAM DISTRIBUTION SERVICES .....293**

**I.2 SERVICE GUIDE.....294**

        I.2.1 Hybrid Broadcast/Unicast Service Guide Discovery .....296

        I.2.2 Hybrid Broadcast/Unicast Service Guide Update and Management .....296

**I.3 HYBRID BROADCAST/UNICAST SERVICE GUIDE INSTANTIATION.....296**

        I.3.1 Scenario of Service Guide with both Broadcast and Unicast Services .....296

        I.3.2 Scenario of Service Guide with Broadcasted Stream Service alternatively available over the Interaction Channel. 301

        I.3.3 Scenario of Service Guide with Broadcasted File Download Service alternatively available over the Interaction Channel.....303

**I.4 EXAMPLE FOR HYBRID BROADCAST/UNICAST SERVICE GUIDE DELIVERY .....304**

# Figures

Figure 1: Structure of Service Guide .....	19
Figure 2: Cardinalities and Reference Direction between Service Guide Fragments .....	20
Figure 3: SG fragment change and its propagation on the different levels of Service Guide, when Split-TOI is not used neither for SGDU nor for SGDD .....	218
Figure 4: Fragment version change and its detection at ALC packet header level of SGDU/SGDD delivery, when Split-TOI is used.....	219
Figure 5: New fragment introduced and its detection at ALC packet header level of SGDU delivery, when Split-TOI is used .....	220
Figure 6: New fragment introduced and its propagation on the different levels of Service Guide, when Split-TOI is not used .....	221
Figure 7: Algorithm for determining a subset of Service Guide.....	223
Figure 8: Example case how preview data supports service-by-service switching .....	235
Figure 9: Example case how preview data supports Service Guide browsing with schedule view.....	236
Figure 10: Example case how preview data supports Service Guide browsing with purchase view .....	237
Figure 11: Example case how preview data supports service preview.....	238
Figure 12: Example case how preview data supports barker .....	239
Figure 13: Example case how preview data supports alternative to blackout .....	240
Figure 14: Discovery of domain name part of the entry point URL with DNS .....	249
Figure 15: Use of ‘Service’, ‘Content’, ‘Schedule’ and ‘Access’ fragment to declare a service with a TV program in two different languages. ....	260
Figure 16: Use of SG to signal service and content protection information .....	261
Figure 17: Service example on a timeline .....	263
Figure 18: Service Guide example for main service with interactivity .....	264
Figure 19: Simple encapsulation of Service Guide fragments in this example .....	264
Figure 20: Service example on a timeline .....	267
Figure 21: Service Guide example for main Service with value-added, protected data .....	267
Figure 22: Simple encapsulation of Service Guide fragments in this example .....	268
Figure 23: Service Guide example .....	269
Figure 24: Simple encapsulation of Service Guide fragments in this example .....	269
Figure 25: Service Guide figure.....	270
Figure 26: Simple encapsulation of Service Guide fragments in this example .....	271
Figure 27: Service Guide figure.....	271
Figure 28: Service Guide example.....	274

Figure 29: Simple encapsulation of Service Guide fragments in this example .....275

Figure 30: Service example on a timeline .....276

Figure 31: Example of Service Guide instantiation in case the delivered files are described in the Service Guide .....276

Figure 32: Service Guide Delivery Descriptor role in Service Guide delivery showing SG fragments grouped by time .....284

Figure 33: General Services set diagram .....293

Figure 34: Typical services set diagram when the interaction channel is used as a fallback delivery method.....294

Figure 35: General Service Guide set diagram .....294

Figure 36: The relation between the service set diagram and Service Guide set diagram .....295

Figure 37: Typical set diagram when the interaction channel extends the service offering.....296

Figure 38: Service example on a timeline .....297

Figure 39: Example of Service Guide instantiation .....298

Figure 40: Service example on a timeline .....301

Figure 41: Example of Service Guide instantiation (fragment in gray is removed compared to the Service Guide instantiation illustrated in Figure 39 , and fragments in green are added compared to the Service Guide instantiation illustrated in Figure 39).....302

Figure 42: Example of Service Guide instantiation for file download service available over both broadcast channel and interaction channel .....304

Figure 43: Services example on a timeline.....305

Figure 44: Example of Service Guide instantiation .....306

## Tables

Table 1: Service Guide Delivery Unit structure .....160

Table 2: Mnemonics used in Table 1 .....160

Table 3: Semantics for Table 1 .....162

Table 4: Semantics of possible values of the <key> "function" .....214

Table 5: Effect of the <key> "function" on associated fragments with given content(s) depending on the presence of the <key> "all" .....215

Table 6: Effect of the <key> "function" on associated fragments with given service(s) depending on the presence of the <key> "all" .....215

Table 7: Uses of 'PreviewData' fragment when referenced by other Service Guide fragments .....234

Table 8: Elements that can be specialized for different languages.....251

Table 9: SGDD Example .....273

Table 10: SGDD Example .....274

Table 11: Cross Reference Table (Informative) .....287

**Table 12: Example of SGDD. Elements and attributes written in *italics* are not instantiated for this example but are include in the table since their absence is also commented in the bullets below.....308**



# 1. Scope

The scope of this specification is the Service Guide functionality of OMA Mobile Broadcast Enabler. The Service Guide functionality is used to provide information about the services and contents available via broadcast channel, interaction channel or both.

The present document defines the Service Guide data model, the interfaces, the delivery and the discovery of the Service Guide functionality (including the declaration of Service Guides within a Service Guide) and the use of PreviewData for services switching, browsing and preview.

## 2. References

### 2.1 Normative References

- [3GPP TS 22.022] “Personalisation of GSM Mobile Equipment (ME); Mobile functionality specification”, 3<sup>rd</sup> Generation Partnership Project, Technical Specification 3GPP TS 22.022 Release 8,  
[URL:http://www.3gpp.org](http://www.3gpp.org)
- [3GPP TS 23.003] “Numbering, Addressing and Identification”, 3<sup>rd</sup> Generation Partnership Project, Technical Specification 3GPP TS 23.003, Release 8,  
[URL:http://www.3gpp.org/ftp/Specs/](http://www.3gpp.org/ftp/Specs/)
- [3GPP TS 26.346] “Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs”, 3<sup>rd</sup> Generation Partnership Project, Technical Specification 3GPP TS 26.346 Release 8,  
[URL:http://www.3gpp.org/ftp/Specs/](http://www.3gpp.org/ftp/Specs/)
- [3GPP TS 31.102] “Characteristics of the Universal Subscriber Identity Module (USIM) application”, 3<sup>rd</sup> Generation Partnership Project, Technical Specification 3GPP TS 31.102, Release 8,  
[URL:http://www.3gpp.org/](http://www.3gpp.org/)
- [3GPP TS 33.220] “Generic Authentication Architecture (GAA), Generic Bootstrapping Architecture”, 3<sup>rd</sup> Generation Partnership Project, Technical Specification 3GPP TS 33.220 Release 8,  
[URL:http://www.3gpp.org](http://www.3gpp.org)
- [3GPP2 C.S0002-E] “Physical Layer Standard for cdma2000 Spread Spectrum Systems”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0002-E, Release E, Version 1.0, September 2009,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 C.S0005-E] “Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems.”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0005-E, Release E, Version 1.0, September 2009,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 C.S0023-D] “Removable User Identity Module for Spread Spectrum Systems”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0023-D, ReleaseD, Version 1.0, June 2009,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 C.S0024-B] “cdma2000 High Rate Packet Data Air Interface Specification”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0024-B, Release B, Version 3.0, September 2009,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 C.S0065-B] “cdma2000 Application on UICC for Spread Spectrum Systems”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0065-B, Release B, Version 1.0, January 2010,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 C.S0068-0] “ME Personalization for cdma2000 Spread Spectrum Systems”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0068-0, Release 0, Version 1.0, May 2006,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [3GPP2 X.S0022-A] “Broadcast and Multicast Service in cdma2000 Wireless IP Network”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 X.S0022-A, Release A, Version 1.0, February 2007,  
[URL:http://www.3gpp2.org/](http://www.3gpp2.org/)
- [BCAST11-Requirements] “Mobile Broadcast Services Requirements”, Open Mobile Alliance™, OMA-RD-BCAST-V1\_1,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [BCAST12-BCMCS-Adaptation] “BCAST Distribution System Adaptation – 3GPP2/BCMCS”, Open Mobile Alliance™, OMA-TS-BCAST\_BCMCS\_Adaptation-V1\_2,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [BCAST12-Distribution] “File and Stream Distribution for Mobile Broadcast Services”, Open Mobile Alliance™, OMA-TS-BCAST\_Distribution-V1\_2,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [BCAST12-DVBH-IPDC-Adaptation] “BCAST Distribution System Adaptation – IPDC over DVB-H”, Open Mobile Alliance™, OMA-TS-BCAST\_DVB\_Adaptation-V1\_2,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)

<b>[BCAST12-DVBNGH-Adaptation]</b>	“BCAST Distribution System Adaptation – DVB Next Generation Handheld”, Open Mobile Alliance™, OMA-TS-BCAST_DVBNGH_Adaptation-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-DVBSH-IPDC-Adaptation]</b>	“BCAST Distribution System Adaptation – IPDC over DVB-SH”, Open Mobile Alliance™, OMA-TS-BCAST_DVBSH_Adaptation-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-FLO-Adaptation]</b>	“BCAST Distribution System Adaptation – Forward Link Only”, Open Mobile Alliance™, OMA-TS-BCAST_FLO_Adaptation-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-MBMS-Adaptation]</b>	“BCAST Distribution System Adaptation – 3GPP/MBMS”, Open Mobile Alliance™, OMA-TS-BCAST_MBMS_Adaptation-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-Schema-sg-be]</b>	“Mobile Broadcast Services – XML Schema for Service Guide Backend Interfaces”, Open Mobile Alliance™, OMA-SUP-XSD_bcast_sg_backend-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-Schema-sg-f]</b>	“Mobile Broadcast Services – XML Schema for Service Guide Fragments”, Open Mobile Alliance™, OMA-SUP-XSD_bcast_sg_fragments-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-Schema-sg-sgdd]</b>	“Mobile Broadcast Services – XML Schema for SGDD”, Open Mobile Alliance™, OMA-SUP-XSD_bcast_sg_sgdd-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-ServContProt]</b>	“Service and Content Protection for Mobile Broadcast Services”, Open Mobile Alliance™, OMA-TS-BCAST_SvcCntProtection-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[BCAST12-Services]</b>	“Mobile Broadcast Services”, Open Mobile Alliance™, OMA-TS-BCAST_Services-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[DM 1.3]</b>	“Enabler Release Definition for OMA Device Management v1.3”, OMA-ERELED-DM-V1_3, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[DRM20-Broadcast-Extensions]</b>	“OMA DRM v2.0 Extensions for Broadcast Support”, Open Mobile Alliance™, OMA-TS-DRM-XBS-V1_2, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[FUMO]</b>	“OMA Enabler Release Definition for Firmware Update Management Object v1.0”, Open Mobile Alliance™, OMA-ERELED-FUMO-V1_0, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[HTML401]</b>	HTML 4.01 Specification, W3C Recommendation 24 December 1999, <a href="http://www.w3.org/TR/html4/">URL:http://www.w3.org/TR/html4/</a> .
<b>[IEEE 802.16-2004]</b>	IEEE 802.16-2004 October 2004, Air Interface for Fixed and Mobile Broadband Wireless Access Systems – Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, August 2004 <a href="http://www.ieee.org">URL:http://www.ieee.org</a>
<b>[IEEE 802.16e-2005]</b>	IEEE 802.16-2005 and IEEE 802.16-2004/Cor 1-2005, Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, February 2006 <a href="http://www.ieee.org">URL:http://www.ieee.org</a>
<b>[IOPPROC]</b>	“OMA Interoperability Policy and Process”, Version 1.13, Open Mobile Alliance™, OMA-IOP-Process-V1_13, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
<b>[ITU-MCC]</b>	“List of Mobile Country or Geographical Area Codes”, ITU-T Telecommunication Standardization Sector of ITU Complement To ITU-T Recommendation E.212 (05/2004), <a href="http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212A-2007-PDF-E.pdf">URL:http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212A-2007-PDF-E.pdf</a> Note: This List will be updated regularly by numbered series of amendments published in ITU Operational Bulletin. For the latest version see: <a href="http://www.itu.int/itu-t/bulletin/annex.html">URL:http://www.itu.int/itu-t/bulletin/annex.html</a>
<b>[MIGFG]</b>	The Moving Image Genre-Form Guide, Library of US Congress, <a href="http://www.loc.gov/r/mopic/miggen.html">URL:http://www.loc.gov/r/mopic/miggen.html</a>

- [OMA MLP] “Mobile Location Protocol”, Open Mobile Alliance™, OMA-TS-MLP-V3\_2,  
[URL:http://www.openmobilealliance.org/](http://www.openmobilealliance.org/)
- [OMNA] Open Mobile Naming Authority, Open Mobile Alliance™,  
[URL:http://www.openmobilealliance.org/tech/omna](http://www.openmobilealliance.org/tech/omna)
- [RFC 1035] “DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION”, P. Mockapetris, November 1987,  
[URL:http://www.ietf.org/rfc/rfc1035.txt](http://www.ietf.org/rfc/rfc1035.txt)
- [RFC 1123] “Requirements for Internet Hosts -- Application and Support”, Internet Engineering Task Force, R. Braden, Editor, October 1989,  
[URL:http://www.ietf.org/rfc/rfc1123.txt](http://www.ietf.org/rfc/rfc1123.txt)
- [RFC 130] “RFC 1305 Network Time Protocol (Version 3) Specification, Implementation and Analysis”, D. Mills, March 1992,  
[URL:http://www.ietf.org/rfc/rfc1305.txt](http://www.ietf.org/rfc/rfc1305.txt)
- [RFC 1952] “GZIP file format specification version 4.3”, P. Deutsch, May 1996,  
[URL:http://www.ietf.org/rfc/rfc1952.txt](http://www.ietf.org/rfc/rfc1952.txt)
- [RFC 2046] “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types”, N. Freed, N. Borenstein, November 1996,  
[URL:http://www.ietf.org/rfc/rfc2046.txt](http://www.ietf.org/rfc/rfc2046.txt)
- [RFC 2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997,  
[URL:http://www.ietf.org/rfc/rfc2119.txt](http://www.ietf.org/rfc/rfc2119.txt)
- [RFC 2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997,  
[URL:http://www.ietf.org/rfc/rfc2234.txt](http://www.ietf.org/rfc/rfc2234.txt)
- [RFC 2246] “The TLS Protocol, Version 1.0”, T. Dierks, C.Allen, January 1999,  
[URL:http://www.ietf.org/rfc/rfc2246.txt](http://www.ietf.org/rfc/rfc2246.txt)
- [RFC 2326] IETF RFC 2326, “Real Time Streaming Protocol (RTSP)”,  
[URL:http://www.ietf.org/rfc/rfc2326.txt](http://www.ietf.org/rfc/rfc2326.txt)
- [RFC 2396] “Uniform Resource Identifiers (URI): Generic Syntax”, T. Berners-Lee et al., August 1998,  
[URL:http://www.ietf.org/rfc/rfc2396.txt](http://www.ietf.org/rfc/rfc2396.txt)
- [RFC 2616] IETF RFC 2616, “Hypertext Transfer Protocol -- HTTP/1.1”,  
[URL:http://www.ietf.org/rfc/rfc2616.txt](http://www.ietf.org/rfc/rfc2616.txt)
- [RFC 2782] IETF RFC 2782, “A DNS RR for specifying the location of services (DNS SRV)”,  
[URL:http://www.ietf.org/rfc/rfc2782.txt](http://www.ietf.org/rfc/rfc2782.txt)
- [RFC 3066] “Tags for the Identification of Languages”, H. Alvestrand, January 2001,  
[URL:http://www.ietf.org/http://www.ietf.org/rfc/rfc3066.txt](http://www.ietf.org/http://www.ietf.org/rfc/rfc3066.txt)
- [RFC 3450] “Asynchronous Layered Coding (ALC) Protocol Instantiation”, M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, December 2002,  
[URL:http://www.ietf.org/rfc/rfc3450.txt](http://www.ietf.org/rfc/rfc3450.txt)
- [RFC 3451] “Layered Coding Transport (LCT) Building Block”, M. Luby, December 2002,  
[URL:http://www.ietf.org/rfc/rfc3451.txt](http://www.ietf.org/rfc/rfc3451.txt)
- [RFC 3695] “Compact Forward Error Correction(FEC) Schemes”, Luby, M. and L. Vicisano,  
[URL:http://www.ietf.org/rfc/rfc3695.txt](http://www.ietf.org/rfc/rfc3695.txt)
- [RFC 3926] “FLUTE - File Delivery over Unidirectional Transport”, T. Paila, October 2004,  
[URL:http://www.ietf.org/rfc/rfc3926.txt](http://www.ietf.org/rfc/rfc3926.txt)
- [RFC 4396] “RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text”, J.Rey, Y. Matsui, IETF RFC 4396, February 2006,  
[URL:http://www.ietf.org/rfc/rfc4396.txt](http://www.ietf.org/rfc/rfc4396.txt)
- [RFC 4566] “SDP – Session Description Protocol”, M. Handley, IETF RFC 4566, July 2006,  
[URL:http://www.ietf.org/rfc/rfc4566.txt](http://www.ietf.org/rfc/rfc4566.txt)
- [RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005,  
[URL:http://www.ietf.org/rfc/rfc4234.txt](http://www.ietf.org/rfc/rfc4234.txt)

[SCOMO]	“OMA Enabler Release Definition for Software Component Management Object v1.0”, Open Mobile Alliance™, OMA-ERELED-SCOMO-V1_0, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
[SCR RULES]	“SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
[SSL30]	“SSL 3.0 Specification”, Netscape Communications, November 1996, <a href="http://wp.netscape.com/eng/ssl3/draft302.txt">URL:http://wp.netscape.com/eng/ssl3/draft302.txt</a>
[TVA-Metadata]	ETSI TS 102 822-3-1 v1.3.1, “Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems (“TV-Anytime”)”, <a href="http://portal.etsi.org">URL:http://portal.etsi.org</a>
[XML]	Extensible Markup Language (XML) 1.1, W3C Recommendation 04 February 2004, edited in place 15 April 2004 <a href="http://www.w3.org/TR/xml11">URL:http://www.w3.org/TR/xml11</a>
[XML-Schema-2]	XML Schema Part 2: Datatypes, Second Edition, W3C Recommendation 28 October 2004, <a href="http://www.w3.org/TR/xmlschema-2/#built-in-datatypes">URL:http://www.w3.org/TR/xmlschema-2/#built-in-datatypes</a>

## 2.2 Informative References

[BCAST11-Architecture]	“Mobile Broadcast Services Architecture”, Open Mobile Alliance™, OMA-AD- BCAST-V1_1, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
[OMADICT]	“Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_9, <a href="http://www.openmobilealliance.org/">URL:http://www.openmobilealliance.org/</a>
[RFC 2048]	“Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures”, N. Freed, J. Klensin, and J. Postel, November 1996, <a href="http://www.ietf.org/rfc/rfc2048.txt">URL:http://www.ietf.org/rfc/rfc2048.txt</a>

## 3. Terminology and Conventions

### 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

The following is the legend used in this specification:

Type: E=Element, A=Attribute, E1=sub-element, E2=sub-element’s sub-element, E[n]=sub-element of element[n-1]

Cardinality: x..y = the number of the presented instance of this element/attribute is in the range from x to y. If x=0, this specific element/attribute is OPTIONAL for network to use, otherwise it is MANDATORY for network to use.

Category: NM = Mandatory for network to support; NO = Optional for network to support; TM = Mandatory for terminal to support; TO = Optional for terminal to support. M = Mandatory to support; O = Optional to support. If an element or attribute has a cardinality greater than zero, it is always classified as M or NM to maintain consistency.

The following relationship applies between elements and their sub-elements respectively attributes:

If an implementation chooses to support an element of category, ...	... it MUST also support all its sub-elements and attributes of category	... it MAY also choose to support any of its sub-element or attribute of category
O	M	O
NO	NM	NO
TO	TM	TO

This is an informative document, which is not intended to provide testable requirements to implementations.

### 3.2 Definitions

Announcement session	Transport sessions which represent the Service Guide Announcement Channel.
Auxiliary Data	Various types of media content, such as text, static image, audio or video clips, etc. used to customize a service or content item, or to provide enhancements or supplementary information to that service or content.
Barker	A content item containing media that is presented to the user in place of a selected content item to which the user is not subscribed.
BCAST Distribution System	A system typically but not necessarily containing the ability to transmit the same IP flow to multiple Terminal devices simultaneously. A BCAST Distribution System (BDS) typically uses techniques that achieve efficient use of radio resources. A BDS consists of Network functionality up to the IP layer and optional Service Distribution/Adaptation functionality above the IP layer. Most BDSs support broadcast/multicast distribution in the network. Some BCAST Distribution Systems have the capability to deliver the IP flows in the network via unicast.
BSMCode	An identifier stored on the BCAST terminal, representing the BSM of the Broadcast Service Provider. For example if the BSMCode is included in the ‘PurchaseChannel’ fragment, it informs the terminal whether the purchase items belonging to that purchase channel may be purchased by the user. BSMCode is presently defined as corresponding to either Smartcard-based terminals or non-Smartcard based terminals.
Cachecast	A non real-time file distribution service, for which the content could consist of audio, audio and video, and/or other types of data. Once the subscriber has subscribed to this service, the content is delivered to the subscriber’s mobile device in the background, during the Distribution Window, transparently to the user. The media is stored on the device and may be accessed by the user during a scheduled availability period referred to as the Presentation Window.

Entry Point	Connectivity related information required by the BCAST Terminal to access service/content, Service Guide, or Service Guide Delivery Descriptor.
GZIP	Abbreviation for GNU zip, a file compression algorithm defined by IETF RFC 1952, “GZIP file format specification version 4.3”.
Preview Data	Various types of media content, such as text, static image, audio clips, video clips, etc, or the combination of the above mentioned media components, used to present the outline of a service, content or purchase item to users. The preview data is described by the ‘PreviewData’ fragment of the Service Guide.
Service Class	A unique name used to identify a particular broadcast service. The service class is a combination of many properties, such as media formats or file types under use, handling of media, interaction with user, etc., used collectively to describe how a particular service works.
Service Guide Announcement Channel	A broadcast distribution channel over which Service Guide Delivery Descriptors carried within announcement sessions, are delivered to the terminal.
Service Guide Delivery Channel	A broadcast distribution channel over which Service Guide Delivery Units are delivered to the terminal.
Service Guide Fragment	An atomic information component of the Service Guide, which can be compressed, encapsulated and transported in the absence of other parts of the Service Guide.

### 3.3 Abbreviations

<b>3GPP</b>	3rd Generation Partnership Project
<b>ALC</b>	Asynchronous Layered Coding
<b>BCAST</b>	Mobile Broadcast Services
<b>BCMCS</b>	Broadcast Multicast Service
<b>BDS</b>	BCAST Distribution System
<b>BSA</b>	BCAST Service Application
<b>BSD/A</b>	BCAST Service Distribution/Adaptation
<b>BSM</b>	BCAST Subscription Management
<b>CID</b>	Content ID
<b>DRM</b>	Digital Rights Management
<b>DVB</b>	Digital Video Broadcast
<b>DVB-H</b>	Digital Video Broadcast – Handheld
<b>DVB-NGH</b>	Digital Video Broadcast – Next Generation Handheld
<b>DVB-SH</b>	Digital Video Broadcast – Satellite to Handheld
<b>DVB-T2</b>	Digital video Broadcast – Second Generation Terrestrial
<b>EXT_CENC</b>	(LCT header) Extension (defining) Content Encoding
<b>EXT_FTI</b>	(LCT header) Extension (defining) FEC Transmission Information
<b>FDT</b>	File Delivery Table
<b>FEC</b>	Forward Error Correction
<b>FLUTE</b>	File Delivery over Unidirectional Transport
<b>GBA</b>	Generic Bootstrapping Architecture
<b>GBA_ME</b>	Mobile Equipment (Terminal) based GBA
<b>GBA_U</b>	GBA with UICC-based enhancements
<b>GZIP</b>	GNU zip
<b>HRPD</b>	High Rate Packet Data

<b>HTTP</b>	Hypertext Transfer Protocol
<b>IP</b>	Internet Protocol
<b>IPDC</b>	IP DataCast
<b>KMS</b>	Key Management System
<b>LCT</b>	Layered Coding Transport
<b>MBMS</b>	Multimedia Broadcast / Multicast Service
<b>MD5</b>	Message Digest (Number) 5
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>MLP</b>	Mobile Location Protocol
<b>MMS</b>	Multimedia Messaging Service
<b>NID</b>	Network Identification
<b>NTP</b>	Network Time Protocol
<b>OMA</b>	Open Mobile Alliance
<b>OTI</b>	Object Transmission Information
<b>PZID</b>	Packet Zone ID
<b>RI</b>	Rights Issuer
<b>RO</b>	Rights Object
<b>RTP</b>	Real-time Transport Protocol
<b>RTSP</b>	Real-Time Streaming Protocol
<b>SDP</b>	Session Description Protocol
<b>SG</b>	Service Guide
<b>SG-A</b>	Service Guide Adaptation
<b>SG-C</b>	Service Guide-Client
<b>SG-D</b>	Service Guide-Distribution
<b>SGDD</b>	Service Guide Delivery Descriptor
<b>SGDU</b>	Service Guide Delivery Unit
<b>SG-G</b>	Service Guide Generation
<b>SID</b>	System Identification
<b>SMIL</b>	Synchronized Multimedia Integration Language
<b>SMS</b>	Short Message Service
<b>TOI</b>	Transport Object Identifier
<b>TSI</b>	Transport Session Identifier
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>USBD</b>	User Service Bundle Description
<b>USD</b>	User Service Description
<b>WAP</b>	Wireless Access Protocol
<b>XML</b>	Extensible Markup Language



## 4. Introduction

This document specifies the Service Guide Function. Service Guide enables the service and content providers to describe the services and content they make available, or offer for subscription or purchase, as Mobile Broadcast services either over Broadcast Channel or over Interaction Channel. It also enables the way to describe how to access the services. From the user perspective the Service Guide can be seen as an entry point to discover the currently available or scheduled services and content and to filter those based on their preferences. Furthermore, the Service Guide provides the entry point to interactive services.

### 4.1 Version 1.0

The Service Guide comprises of data model that models the services, schedules, content, related purchase and provisioning data, access and interactivity data in terms of Service Guide fragments. Further, this specification defines methods for initial discovery of Service Guide and the declaration of Service Guide fragments through Service Guide Delivery Descriptors that are sent over Service Guide Announcement Channel. The delivery methods are specified for Service Guide fragment delivery over Service Guide Delivery Channel, both over the Broadcast Channel as well as over the Interactive Channel. To manage the Service Guide data, the specification defines update and management methods. Finally, this document specifies the relevant back-end interfaces for Service Guide.

### 4.2 Version 1.1

In this version, some features have been enhanced (e.g., unicast SG query) and the following main additional features are introduced:

- Support signaling for broadcast terminal and smartcard provisioning
- Support signaling for Audience Measurement
- Support Rich Media Solution (RMS) signaling for Service Guide presentation, Streaming, File delivery and Preview Data
- Support location based request of SG
- Support signaling of hybrid SG and hybrid Services
- Support Service Guide Discovery using DNS
- Support signaling for user generated content
- Support enhanced location broadcast area
- Support signaling for user defined bundle.
- Support signaling for WiMAX Unicast, DVB-SH, and FLO-IP

### 4.3 Version 1.2

BCAST 1.2 aims to provide BDS adaptation for the IP profile of the Digital Video Broadcasting – Next Generation Handheld (DVB-NGH) distribution system, Digital Video Broadcasting – Second Generation Terrestrial (DVB-T2) and its T2-Lite profile tailored for mobile reception. The required Service Guide information to support these two new BDSs are done.

## 5. Service Guide

### 5.1 Service Guide Data Model

#### 5.1.1 Service Guide Structure and Fragmentation

The structure of the Service Guide data model is illustrated in the Figure 1 below. The basic assumption is that the data is represented as XML fragments. Each XML fragment is considered as a separate well-formed XML document [XML]. The XML text declaration MAY be omitted. In such a case, the terminal SHALL assume the following default XML text declaration to ensure well-formedness:

```
<?xml version="1.1"?>
```

The namespaces used in a fragment SHOULD be declared in the fragment according to XML rules [XML]. If no namespace is declared, the terminal SHALL assume that the default namespace of the fragment is “urn:oma:xml:bcast:sg:fragments:1.2”.

The meaning of the cardinalities shown in the Figure 1 is the following: One instantiation of Fragment A as in Figure 2 references c to d instantiations of Fragment B. If c=d, d is omitted. Thus, if  $c > 0$  and Fragment A exists, at least c instantiation of Fragment B must also exist, but at most d instantiations of Fragment B may exist. Vice versa, one instantiation of Fragment B is referenced by a to b instantiations of Fragment A. If a=b, b is omitted. The arrow connection from Fragment A pointing to Fragment B indicates that Fragment A contains the reference to Fragment B.

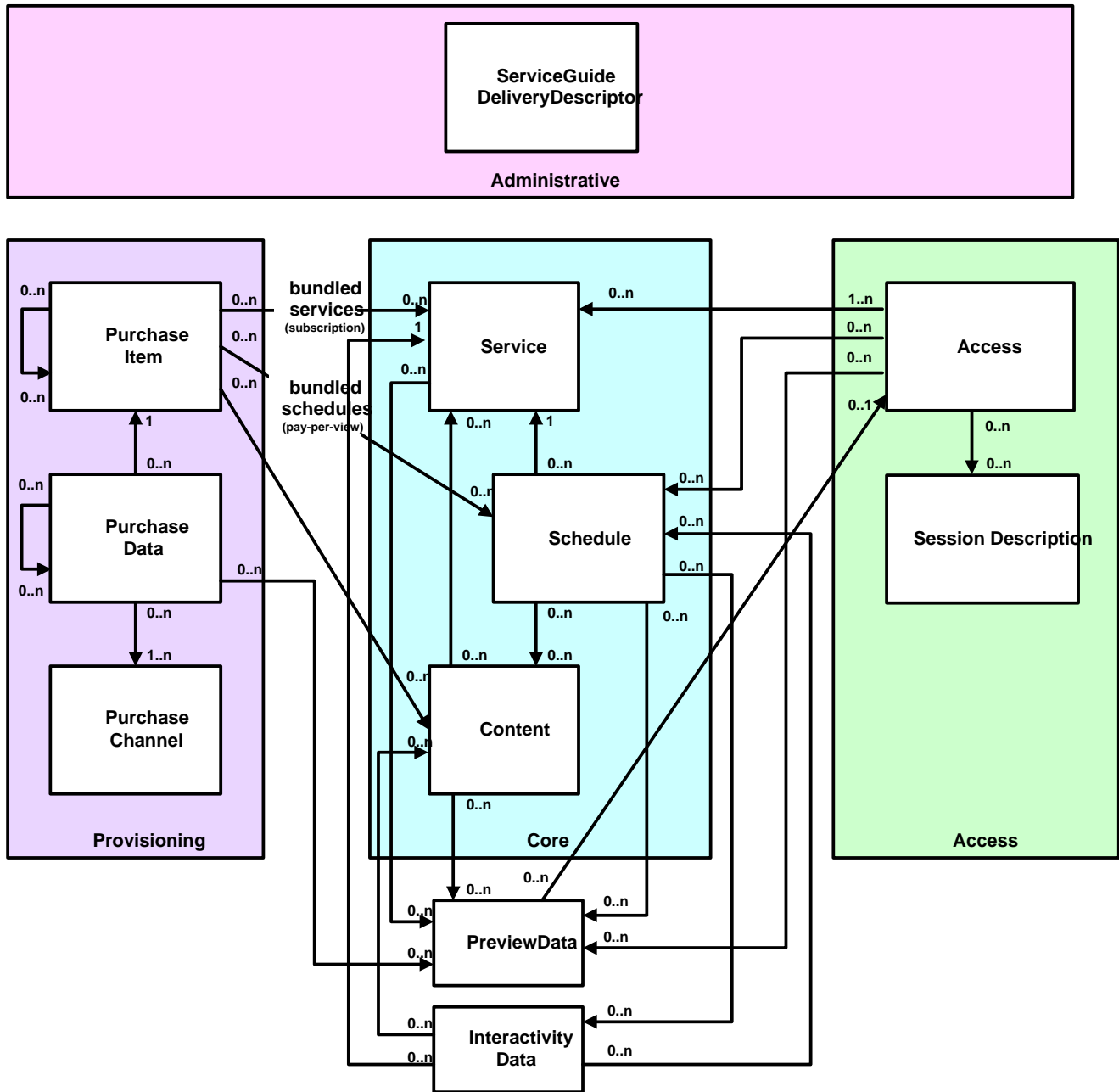
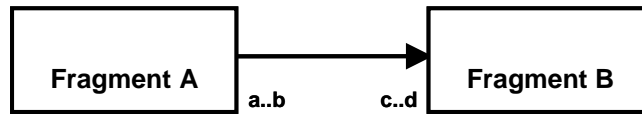


Figure 1: Structure of Service Guide

Any given ‘PurchaseItem’ fragment SHALL only be able to reference a single type among the following fragments: ‘Service’, ‘Schedule’, ‘Content’, or another ‘PurchaseItem’. ‘Access’ fragment SHALL have a link to either ‘Service’ fragment or ‘Schedule’ fragment.

As shown in Figure 1, all of the connection arrows between Service Guide fragments are uni-directional, specially, there are two pairs of opposite uni-directional arrows: one pair is between ‘Schedule’ fragment and ‘InteractivityData’ fragment, and the other pair is between ‘Access’ fragment and ‘PreviewData’ fragment. The reference arrow from ‘Schedule’ fragment to ‘InteractivityData’ fragment declares the distribution schedule of the interactive media documents carried in a file stream (referenced by the ‘InteractivityData’ fragment); the reference arrow from ‘InteractivityData’ fragment to ‘Schedule’ fragment declares which ‘Schedule’ Fragment this ‘InteractivityData’ Fragment is associated with. The reference arrow from

'Access' fragment to 'PreviewData' fragment indicates the service-by-service switching preview information for the access; the reference arrow from 'PreviewData' fragment to 'Access' fragment declares how the preview data can be accessed.



**Figure 2: Cardinalities and Reference Direction between Service Guide Fragments**

The semantics of the elements in the model as defined as follows:

### *Service*

The 'Service' fragment describes at an aggregate level the content items which comprise a broadcast service. The service may be delivered to the user using multiple means of access, for example, the broadcast channel and the interactive channel. The service may be targeted at a certain user group or geographical area. Depending on the type of the service it may have interactive part(s), broadcast-only part(s), or both.

Further, the service may include components not directly related to the content but to the functionality of the service – such as purchasing or subscription information. As the part of the Service Guide, the 'Service' fragment forms a central hub referenced by the other fragments including 'Access', 'Schedule', 'Content' and 'PurchaseItem' fragments. In addition to that, the 'Service' fragment may reference 'PreviewData' fragment. It may be referenced by none or several of each of these fragments.

Together with the associated fragments the terminal may determine the details associated with the service at any point of time. These details may be summarized into a user-friendly display, for example, of what, how and when the associated content may be consumed and at what cost

### *Schedule*

The 'Schedule' fragment defines the timeframes in which associated content items are available for streaming, downloading and/or rendering. This fragment always references the 'Service' fragment. If it also references one or more 'Content' fragments or 'InteractivityData' fragments, then it defines the valid distribution and/or presentation timeframe of those content items belonging to the service, or the valid distribution timeframe and the automatic activation time of the InteractivityMediaDocuments associated with the service. On the other hand, if the 'Schedule' fragment does not reference any 'Content' fragment(s) or 'InteractivityData' fragment(s), then it defines the timeframe of the service availability which is unbounded.

### *Content*

The 'Content' fragment gives a detailed description of a specific content item. In addition to defining a type, description and language of the content, it may provide information about the targeted user group or geographical area, as well as genre and parental rating.

The 'Content' fragment may be referenced by Schedule, PurchaseItem or 'InteractivityData' fragment. It may reference 'PreviewData' fragment or 'Service' fragment.

### *Access*

The 'Access' fragment describes how the service may be accessed during the lifespan of the service. This fragment contains or references Session Description information and indicates the delivery method. One or more 'Access' fragments may reference a 'Service' fragment, offering alternative ways for accessing or interacting with the associated service.

For the Terminal, the 'Access' fragment provides information on what capabilities are required from the terminal to receive and render the service. The 'Access' fragment provides Session Description parameters either in the form of inline text, or through a pointer in the form of a URI to a separate Session Description. Session Description information may be delivered over either the broadcast channel or the interaction channel.

### *SessionDescription*

The 'SessionDescription' is a Service Guide fragment which provides the session information for access to a service or content item. Further, the Session Description may provide auxiliary description information, used for associated delivery procedures.

The Session Description information is provided using either syntax of SDP in text format, or through a 3GPP MBMS User Service Bundle Description [3GPP TS 26.346] (USBD).

Auxiliary description information is provided in XML format and contains an Associated Delivery Description as specified in [BCAST12-Distribution].

Note that in case SDP syntax is used, an alternative way to deliver the Session Description is by encapsulating the SDP in text format in 'Access' fragment.

Note that Session Description as a concept may be used both for Service Guide delivery itself as well as for the content sessions.

### *PurchaseItem*

The 'PurchaseItem' fragment represents a group of one or more services (i.e. a service bundle) or one or more content items, offered to the end user for free, for subscription and/or purchase.

This fragment can be referenced by 'PurchaseData' fragment(s) offering more information on different service bundles. The 'PurchaseItem' fragment may be also associated with:

- a 'Service' fragment to enable bundled services subscription and/or,
- a 'Schedule' fragment to enable consuming a certain service or content in a certain timeframe (pay-per-view functionality) and/or,
- a 'Content' fragment to enable purchasing a single content file related to a service.
- other 'PurchaseItem' fragments to enable bundling of purchase items

### *PurchaseData*

The main function of the 'PurchaseData' fragment is to express all the available pricing information about the associated purchase item.

The 'PurchaseData' fragment collects the information about one or several purchase channels and may be associated with PreviewData specific to a certain service or service bundle. It carries information about pricing of a service, a service bundle, or, a content item. Also, information about promotional activities may be included in this fragment.

When a 'PurchaseData' fragment is used, this may initiate a second follow-on (autosubscribe) purchase. Each follow-on 'PurchaseData' fragment will point to the originally purchased 'PurchaseData' fragment. If the follow-on purchase is not free, the terminal must prompt the user for acceptance of the purchase. To create an automatically subscribed 'PurchaseData', the fragment may point to itself. Service providers may use this feature e.g. for automatic free-to-air encrypted services, for cross-promotions, and to provision the terminal for higher-layer BCAST applications such as DCD, DM, etc.

### *PurchaseChannel*

The 'PurchaseChannel' fragment carries the information about the entity from which purchase of access and/or content rights for a certain service, service bundle or content item may be obtained, as defined in the 'PurchaseData' fragment. The purchase channel is associated with one or more Broadcast Subscription Managements (BSMs). The terminal is only permitted to access a particular purchase channel if it is affiliated with a BSM that is also associated with that purchase channel.

Multiple purchase channels may be associated to one 'PurchaseData' fragment. A certain end-user can have a "preferred" purchase channel (e.g. his/her mobile operator) to which all purchase requests should be directed. The preferred purchase channel may even be the only channel that an end-user is allowed to use.

### *PreviewData*

'PreviewData' fragment contains information that is used by the terminal to present the service or content outline to users, so that the users can have a general idea of what the service or content is about. 'PreviewData' fragment can include simple texts, static images (for example, logo), short video clips, or even reference to another service which could be a low bit rate version for the main service. 'Service', 'Content', 'PurchaseData', 'Access' and 'Schedule' fragments may reference 'PreviewData' fragment

### *InteractivityData*

The InteractivityData contains information that is used by the terminal to offer interactive services to the user, which is associated with the broadcast content. These interactive services enable users to e.g. vote during TV shows or to obtain content related to the broadcast content. 'InteractivityData' fragment points to one or many 'InteractivityMedia' documents that include xhtml files, static images, email template, SMS template, MMS template documents, etc. The 'InteractivityData' fragment may reference the 'Service', 'Content' and 'Schedule' fragments, and may be referenced by the 'Schedule' fragment.

### *ServiceGuideDeliveryDescriptor*

The ServiceGuideDeliveryDescriptor is transported on the Service Guide Announcement Channel, and informs the terminal the availability, metadata and grouping of the fragments of the Service Guide in the Service Guide discovery process (see section 6.1.1). A SGDD allows quick identification of the Service Guide fragments that are either cached in the terminal or being transmitted. For that reason, the SGDD is preferably repeated if distributed over broadcast channel. The SGDD also provides the grouping of related Service Guide fragments and thus a means to determine completeness of such group.

The ServiceGuideDeliveryDescriptor is especially useful if the terminal moves from one service coverage area to another. In this case, the ServiceGuideDeliveryDescriptor can be used to quickly check which of the Service Guide fragments that have been received in the previous service coverage area are still valid in the current service coverage area, and therefore don't have to be re-parsed and re-processed.

## **5.1.2 Service Guide Elements, Attributes and Meta Data**

This section defines the Service Guide metadata, including elements and attributes of every Service Guide fragment and the type, cardinality, category, description and data type of each element and attribute.

In 'Service' fragment, 'Content' fragment, 'PurchaseItem' fragment, 'PurchaseData' fragment and 'PurchaseChannel' fragment, there are some elements or attributes which are particularly intended for end-user as program guide. This kind of metadata are grouped and highlighted between the "Start of program guide" and "End of program guide" cells in the above mentioned fragments. The program guide elements or attributes SHOULD be used for presentation purposes, and MAY be used for terminal filtering, search, sort, etc. Other elements or attributes can be used for presentation only, for terminal interpretation only, or for both.

Every SG fragments specified in this section may have zero, one or more than one instances in a Service Guide.

The tables specifying each Service Guide fragment in the following sections use various data types. The data types SHALL conform to [XML-Schema-2].

Terminals with interaction channel (e.g. UMTS, GPRS) SHALL support all SG fragments. Terminals with no return channel MAY NOT support the 'InteractivityData' fragment, but SHALL support all the other SG fragments.

Network SHOULD support and MAY use all SG fragments.

The Service Guide Data Model is formally reflected as XML Schema in [BCAST12-Schema-sg-f].

The terminal SHALL NOT offer an interface that allows modification or removal of any information contained in the service guide. As an exception it MAY be possible to remove whole fragments and the information contained therein, however then it SHALL NOT be possible to access or render the service or program described therein anymore. It MAY be possible to add fragments to the service guide.\*

\*The normative statements in this paragraph are not intended to be a part of the SCR tables.

### 5.1.2.1 Service

A service represents a bundle of content items, which forms a logical group to the end-user. An example would be a TV channel, composed of several TV shows. A ‘Service’ fragment contains the metadata describing the Mobile Broadcast service. It is possible that the same metadata (i.e., attributes and elements) exist in the ‘Content’ fragment(s) associated with that ‘Service’ fragment. In that situation, for the following elements: ‘ParentalRating’, ‘TargetUserProfile’, ‘Genre’ and ‘BroadcastArea’, the values defined in ‘Content’ fragment take precedence over those in ‘Service’ fragment.

Name	Type	Category	Cardinality	Description	Data Type
Service	E			‘Service’ fragment Contains the following attributes: id version validFrom validTo globalServiceID weight hidden baseCID emergency UDBAllowed amAllowed Contains the following elements: ProtectionKeyID ServiceType Name Description AudioLanguage TextLanguage ParentalRating TargetUserProfile Genre Extension PreviewDataReference BroadcastArea TermsOfUse Popularity PrivateExt	
id	A	NM/ TM	1	ID of the ‘Service’ fragment. The value of this attribute SHALL be globally unique.”	anyURI
version	A	NM/ TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no ‘validFrom’ attribute is given.	unsignedInt

validFrom	A	NM/ TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future.	unsignedInt
globalServiceID	A	NM/ TM	0..1	The globally unique identifier identifying the service this 'Service' fragment describes.	anyURI
weight	A	NM/ TM	0..1	Intended order of display of this service relative to other services as presented to the end user. The order of display is by increasing weight value (i.e., service with lowest weight is displayed first). Default: 65535 User preference, if available, SHALL override the weight.	unsignedShort
hidden	A	NM/ TM	0..1	If this flag is present and set to TRUE, the Program Guide for this service SHOULD NOT be displayed to the user.	boolean
baseCID	A	NO/ TO	0..1	For the DRM Profile, part of the Service or Program CID used to identify the corresponding asset within a OMA DRM 2.0 Rights Object. The Service or Program CID is obtained from the BaseCID as described in [BCAST12-ServContProt] section 5.5.1. This element is only Mandatory to support for the network and terminal in case the DRM Profile is supported [BCAST12-ServContProt]. Note: for uniqueness of the baseCID see Appendix H.	string
emergency	A	NO/ TO	0..1	When assigned with value 'true', specifies that this service is a service of emergency nature. That also means that all content items belonging to this service are contents of emergency nature. This attribute can be used for presentation purposes to users. It is RECOMMENDED that the Terminal processes the reception of the services or content of emergency nature with high priority, and highlights their availability to user. How to order the emergency service or content is out of the scope of the specification. The default value of this attribute is 'false'.	boolean
UDAllowed	A	NO TO	0..1	Represents whether if this Service can be used in User Defined Bundle subscriptions.	boolean



amAllowed	A	NO/TO	0..1	<p>This attribute indicates whether Audience Measurement using the Audience Measurement solutions specified in section 5.20 of [BCAST12-Services] is allowed or not for this service. If the value of amAllowed is true or the attribute is not present in the 'Service' fragment then the measurement is allowed. If the value of amAllowed is false, Measurement is not allowed.</p> <p>Note : Usage of amAllowed is further described in section 5.20 of [BCAST12-Services]</p>	Boolean
ProtectionKeyID	E1	NO/TO	0..N	<p>Key identifier needed to access a protected service. This information allows the terminal to determine whether or not it has the correct key material to access service(s) within a PurchaseItem.</p> <p>In a scenario where this fragment is shared among multiple service providers, different key identifiers from the different service providers to access this specific protected service/content may be mixed in this element and the terminal SHOULD be able to sort out the key identifiers associated with the terminal's affiliated service provider based on the Key Domain ID. How this is used is out of scope and is left to implementation.</p> <p>The network and terminal SHALL support this element in case the Smartcard Profile is supported [BCAST12-ServContProt].</p> <p>The protection key identifiers to access a specific service or content item SHALL only be signalled in one of the following fragment types: 'Service', 'Content', 'PurchaseItem', 'PurchaseData' or 'Access' fragments, but not mixed.</p> <p>Contains the following attribute: type</p>	base64Binary

type	A	NM/TM	1	<p>Type of ProtectionKeyID:</p> <p>0: ProtectionKeyID is the 5-byte long concatenation of the Key Domain ID with the Key group part of the SEK/PEK ID, where both values are as used in the Smartcard Profile [BCAST12-ServContProt].</p> <p>The Key number part SHALL NOT be provided.</p> <p>The Terminal MAY use the Key Domain ID and Key group part of the ProtectionKeyID to determine whether it already has the SEK applicable to the related service. The Terminal MAY use this information to indicate to the user which services can currently be accessed. The Terminal SHALL not use the SEK ID in the ProtectionKeyID to request a missing SEK. It is possible for the Terminal to request missing SEK based on the information from the secure function after the STKM decryption has been failed.</p> <p>1-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte
------	---	-------	---	---	--------------

ServiceType	E1	NM/ TM	0..N	<p>Type of the service. Allowed values are:</p> <ul style="list-style-type: none"> <li>0 – unspecified</li> <li>1 – Basic TV</li> <li>2 – Basic Radio</li> <li>3 – RI services</li> <li>4 – Cachecast</li> <li>5 – File download services</li> <li>6 – Software management services</li> <li>7 – Notification</li> <li>8 – Service Guide</li> <li>9 – Terminal Provisioning services</li> <li>10 – Auxiliary Data</li> <li>11 – Streaming on demand</li> <li>12 – File download on demand</li> <li>13 – Smartcard Provisioning services</li> <li>14-127 reserved for future use</li> <li>128-223 reserved for other OMA Enablers</li> <li>224-255 reserved for proprietary use</li> </ul> <p>The mixed service types SHALL be indicated by the presence of multiple instances of ServiceType (for example, for mixed Basic TV and Cachecast, two instances of ServiceType, with values 1 and 4 are present for this ‘Service’ fragment.</p> <p>This element SHALL be processed by the terminal strictly for rendering to the user for example as a textual indicator, an icon, or graphic representation for the service. However, ‘ServiceType’ with value of 3, 9 and 13 SHALL NOT be rendered and their existence SHOULD NOT be displayed to the user. If ‘ServiceType’ is 10 and this is a BCAST 1.0 fragment, then the associated Program Guide portion of this fragment SHOULD NOT be displayed.</p> <p>With value 6, i.e. software management services, users can select the desired software components (Eg. desktop theme, ringtone, SG navigator update) to download over broadcast channel or interaction channel. The software components provided by this software management service are described by ‘Content’ fragments which belong to this ‘Service’ fragment. It is not expected that terminals are able to automatically select and download software components using this type of service.</p> <p>If the terminal supports the ‘AuxDataTrigger’ notification type, and it supports auxiliary data download/caching for subsequent insertion/rendering to users (as described in [BCAST12-Services]), then the content items belonging to this service SHALL be downloaded and selectively cached by the terminal in accordance to the ‘AuxDataTrigger’ element of &lt;type&gt; = 0 (i.e. download trigger) in the Notification message (Section 5.14.3 of [BCAST12-Services]).</p>	unsigned Byte
-------------	----	-----------	------	--	------------------

© 2017 Open Mobile Alliance All Rights Reserved. Used with the permission of the Open Mobile Alliance under the terms as stated in this document.

Services which "ServiceType" equals to "11" or "12" provide "On demand" contents over the Interaction Channel. They do not relate to e.g.

				<p><b>Start of program guide</b></p> <p>The program guide elements of this fragment are grouped between the Start of program guide and end of program guide cells in this fragment.</p> <p>The program guide elements are for user interpretation. This enables the content creator to provide user readable information about the service. The terminal SHOULD use all declared program guide elements in this fragment for presentation to the end-user. The terminal MAY offer search, sort etc functionalities.</p> <p>The Program Guide consists of the following elements:</p> <p>Name Description AudioLanguage TextLanguage ParentalRating TargetUserProfile Genre Extension</p>	
Name	E1	NM/ TM	1..N	Name of the Service, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
Description	E1	NM/ TM	0..N	Description, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
AudioLanguage	E1	NM/ TM	0..N	<p>This element declares for the end users that this service is available with an audio track corresponding to the language represented by the value of this element.</p> <p>The textual value of this element can be made available for the end users in different languages. In such a case the language used to represent the value of this element is signalled using the built-in XML attribute 'xml:lang'. See section 7, Multi-language support.</p> <p>Contains the following attribute: languageSDPtag</p>	string
languageSDPtag	A	NM/ TO	1	<p>Identifier of the audio language described by the parent 'AudioLanguage' element as used in the media sections describing the audio track in a Session Description.</p> <ul style="list-style-type: none"> <li>The 'languageSDPtag' SHALL be formatted according to the rules of [RFC 3066], for the described language.</li> <li>Each 'AudioLanguage' element declaring the same audio stream SHALL have the same value of the 'languageSDPtag'.</li> </ul>	string

TextLanguage	E1	NM/ TM	0..N	<p>This element declares for the end user that the textual components of this service are available in the language represented by the value of this element. The textual components can be, for instance, a caption or a sub-title track.</p> <p>The textual value of this element can be made available for the end users in different languages. In such a case the language used to represent the value of this element is signalled using the built-in XML attribute 'xml:lang'. See section 7 Multi-language support.</p> <p>The same rules and constraints as specified for the element 'AudioLanguage' of assigning and interpreting the attributes 'languageSDPtag' and 'xml:lang' SHALL be applied for this element also.</p> <p>Contains the following attribute: languageSDPtag</p>	string
languageSDPtag	A	NM/TO	1	Identifier of the text language described by the parent 'TextLanguage' element as used in the media sections describing the textual track in a Session Description.	string
ParentalRating	E1	NM/ TM	0..N	<p>The ParentalRating element defines criteria parents might use to determine whether the associated item is suitable for access by children, defined according to the regulatory requirements of the service area.</p> <p>The terminal SHALL support 'ParentalRating' being a free string, and the terminal MAY support the structured way to express the parental rating level by using the 'ratingSystem' and 'ratingValueName' attributes as defined below.</p> <p>Contains the following attributes: ratingSystem ratingValueName</p>	string

ratingSystem	A	NO/ TM	0..1	<p>Specifies the parental rating system in use, in which context the value of the 'ParentalRating' element is semantically defined. This allows terminals to identify the rating system in use in a non-ambiguous manner and act appropriately. This attribute SHALL be instantiated when a rating system is used. Absence of this attribute means that no rating system is used. (i.e. the value of the 'ParentalRating' element is to be interpreted as a free string).</p> <p>If this attribute is instantiated:</p> <ul style="list-style-type: none"> <li>– The value of this attribute SHALL be one of the 'rating_type' values as listed in the OMA BCAST Parental Rating System Registry at [OMNA].</li> <li>– The 'ParentalRating' element SHALL contain the string representation of a number that is a valid 'rating_value' in this particular rating system.</li> <li>– This attribute MAY contain the value '10' (OMA BCAST generic rating scheme), allowing to define a rating value in a non-registered parental rating system. In such case, the 'ParentalRating' element SHALL contain the string representation of a number between 1 and 255, 1 being the least and 255 being the most restrictive rating value. As these values are generic, the human-readable label of that rating value SHALL be signalled in the attribute 'ratingValueName'.</li> </ul>	unsignedByte
ratingValueName	A	NO/TM	0..1	<p>The human-readable name of the rating value given by this ParentalRating element. This attribute SHALL be present in case the 'ratingSystem' attribute contains the value '10'.</p>	string

TargetUserProfile	E1	NO/TO	0..N	<p>Profile attributes of the users whom the service is targeting at. The detailed personal attribute names and the corresponding values are specified by attributes of 'attributeName' and 'attributeValue'. Amongst the possible profile attribute names are age, gender, occupation, etc. (subject to national/local rules &amp; regulations, if present and as applicable regarding use of personal profiling information and personal data privacy).</p> <p>The extensible list of 'attributeName' and 'attributeValue' pairs for a particular service enables end user profile filtering and end user preference filtering of broadcast services. The terminal SHOULD be able to support 'TargetUserProfile' element. The terminal behavior for interpreting and acting upon 'TargetUserProfile' is out of the scope.</p> <p>It is RECOMMENDED that use of 'TargetUserProfile' element is an "opt-in" capability for users. Terminal settings SHOULD allow users to configure whether to input their personal profile or preference and whether to allow broadcast service to be automatically filtered based on the users' personal attributes without users' request.</p> <p>Contains the following attributes:                  attributeName                  attributeValue</p>	
attributeName	A	NM/TM	1	Profile attribute name	string
attributeValue	A	NM/TM	1	Profile attribute value	string

Genre	E1	NM/ TM	0..N	<p>Classification of service associated with characteristic form (e.g. comedy, drama). The OMA BCAST Service Guide allows describing the format of the Genre element in the Service Guide in two ways:</p> <ul style="list-style-type: none"> <li>• The first way is to use a free string</li> <li>• The second way is to use the “href” attributes of the Genre element to convey the information in the form of a controlled vocabulary (classification scheme as defined in [TVA-Metadata] or classification list as defined in [MIGFG]).</li> </ul> <p>The built-in XML attribute xml:lang MAY be used with this element to express the language. The Network MAY instantiate several different sets of ‘Genre’ element, using it as a free string or with a ‘href’ attribute. The Network SHALL ensure the different sets have equivalent and non-conflicting meaning, and the terminal SHALL select one of the sets to interpret for the end-user. Contains the following attributes: type href</p>	string
type	A	NO/ TO	0..1	<p>This attribute signals the level of this ‘Genre’ element. The following values are allowed: “main” “secondary” “other”</p>	string



href	A	NO/ TO	0..1	<p>This attribute signals the controlled vocabulary used for this ‘Genre’ element.</p> <p>If this attribute is supported, the following applies to the support and use of classification schemes according to [TVA-Metadate]:</p> <ul style="list-style-type: none"> <li>for values of the ‘type’ attribute equal to "main" or "secondary", the terminal MAY support levels 1-4 of the TV Anytime ContentCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:ContentCS:2005 as defined in Annex A.8 of [TVA-Metadate]</li> <li>for a value of the ‘type’ attribute equal to "other", the terminal MAY support levels 1-3 of the TV Anytime IntendedAudienceCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:IntendedAudienceCS:2005 as defined in Annex A.11 of [TVA-Metadate]. When the IntendedAudienceCS is provided simultaneously with an instantiation of the ‘TargetUserProfile’, the two SHALL have equivalent meaning.</li> <li>The network SHALL use the following URI syntax to signal terms from classification schemes: &lt;classificationSchemeURI&gt; “:” &lt;termID&gt;</li> <li>If this attribute is instantiated by the network, the element ‘Genre’ SHALL be an empty string and the xml:lang attribute SHALL NOT be used.</li> </ul> <p>If this attribute is supported, the following applies to the support and use of the classification from [MIGFG]:</p> <ul style="list-style-type: none"> <li>This classification SHALL be signalled with the URI “http://www.loc.gov/rr/mopic/miggen.html”</li> <li>The string value carried in the ‘Genre’ element SHALL be used to convey the actual value of the classification as given in [MIGFG]</li> <li>The Network MAY use the values “main” and “secondary” of the ‘type’ attribute so as to provide an ordering of two classifications applying to the same Service.</li> </ul> <p>Other Classification Schemes MAY be signalled with the 'href' attribute, however how they are used is out of scope of this specification.</p> <p>If this attribute is not instantiated, the ‘Genre’ element SHALL be a free string.</p>	anyURI
------	---	-----------	------	---	--------

Extension	E1	NM/ TM	0..N	Additional information related to this fragment. Contains the following attribute: url Contains the following element: Description	
url	A	NM/ TM	1	URL containing additional information related to this fragment.	anyURI
Description	E2	NM/ TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
<b>End of program guide</b>					
PreviewData Reference	E1	NM/ TM	0..N	Reference to the 'PreviewData' fragment which specifies the preview data (Eg. picture, video clip, or low-bit rate stream) associated with this service:  It is possible that there are more than one 'PreviewDataReference' instances associated with the same fragment, in which case, the values of 'usage' attributes of these 'PreviewDataReference' instances SHALL be mutually exclusive. Contains the following attributes: idRef usage	
idRef	A	NM/ TM	1	Identification of the 'PreviewData' fragment which this fragment is associated with.	anyURI
usage	A	NM/ TM	1	Specifies the usage of the associated preview data. Possible values: 0. unspecified 1. Service-by-Service Switching 2. Service Guide Browsing 3. Service Preview 4. Barker 5. Alternative to blackout 6-127. reserved for future use 128-255. reserved for proprietary use The explanation and limitation on the above preview data usages is specified in section 5.7.	unsignedByte

BroadcastArea	E1	NO/TO	0..1	<p>Broadcast area to include location information for BCAST contents.</p> <p>Contains the following attribute:</p> <p>Polarity</p> <p>filteringTime</p> <p>Contains the following elements:</p> <p>TargetArea</p> <p>lev_conf</p> <p>LocationFilter</p> <p>BCAST 1.1 terminals that are location-capable SHALL use only LocationFilter, polarity and filteringTime if present. Otherwise if the LocationFilter element is not present, BCAST 1.1 terminals that are location-capable SHALL use polarity, filteringTime, TargetArea and lev_conf to compute location presence.</p>	
polarity	A	NO/TO	0..1	<p>Indication of whether the associated target area is intended for positive or negative terminal reception of the service.</p> <p>If polarity = true or 1, this indicates the associated service is intended for reception by terminals located within the corresponding geographical area. (Default)</p> <p>If polarity = false or 0, this indicates the associated service is not intended for reception by terminals located within the corresponding geographical area.</p>	boolean
filteringTime	A	NO/TO	0..1	<p>For cachecast service/content, this attribute indicates whether the filtering against target area occurs at the service-guide fragment broadcast time, content download time, or the content rendering time. The terminal SHOULD obtain its current location information before performing filtering.</p> <p>0 – the download time specified by the DistributionWindow in the corresponding Schedule fragment.</p> <p>1 – the rendering time specified by the PresentationWindow in the corresponding Schedule fragment.</p> <p>2 – the location-based filter is applied as soon as this fragment is received by the terminal.</p> <p>3-255 – reserved for future use.</p> <p>Note that when this attribute is present, BCAST 1.1 terminals SHOULD perform filtering at the time indicated by the "filteringTime" value whereas BCAST 1.0 terminals ignore this attribute. The service provider should create their content filters with this limitation in mind.</p>	unsignedByte

TargetArea	E2	NO/ TM	0..N	The target area to distribute contents (as specified in the [OMA MLP] with modifications) Contains the following elements: shape cc mcc name_area ZipCode CellTargetArea Only one of the above six elements SHALL be instantiated at the same time. Implementation in XML schema using <choice>.	
shape	E3	NO/ TM	0..1	Shapes used to represent a geographic area that describes (as specified in the [OMA MLP])	
cc	E3	NO/ TM	0..1	Country code, 1-3 digits e.g. 355 for Albania (as specified in the [OMA MLP])	unsignedShort
mcc	E3	NO/ TM	0..1	Mobile country code, 3 digits e.g. 276 for Albania (as specified in [ITU-MCC], aligned with [OMA MLP])	string of three digits
name_area	E3	NO/ TM	0..N	Geopolitical name of area such as 'Seoul' (as specified in the [OMA MLP]. The instances of 'name_area' element differ only in language. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
ZipCode	E3	NO/ TM	0..1	Zip code	string
CellTargetArea	E3	NO/ TM	0..1	The target area to distribute content specified by the BDS specific service coverage area or minimum transmit area Contains the following attribute: type Contains the following element: CellArea	

type	A	NM/ TM	1	<p>Allowed values are:</p> <ul style="list-style-type: none"> <li>0 – Unspecified</li> <li>1 – 3GPP Cell Global Identifier as defined in [3GPP TS 23.003]</li> <li>2 – 3GPP Routing Area Identifier (RAI) as defined in [3GPP TS 23.003]</li> <li>3 – 3GPP Location Area Identifier (LAI) as defined in [3GPP TS 23.003]</li> <li>4 – 3GPP Service Area Identifier (SAI) as defined in [3GPP TS 23.003]</li> <li>5 – 3GPP MBMS Service Area Identity (MBMS SAI) as defined in [3GPP TS 23.003]</li> <li>6 – 3GPP2 Subnet ID as defined in [3GPP2 X.S0022-A]</li> <li>7 – 3GPP2 SID as defined in [3GPP2 C.S0005-E]</li> <li>8 – 3GPP2 SID+NID as defined in [3GPP2 C.S0005-E]</li> <li>9 – 3GPP2 SID+NID+PZID as defined in [3GPP2 C.S0005-E]</li> <li>10 – 3GPP2 SID+PZID as defined in [3GPP2 C.S0005-E]</li> <li>11 – DVB-H Cell ID (specified in section 6.3.4.1 of [BCAST12-DVBH-IPDC-Adaptation] )</li> <li>12 – DVB-SH Cell ID (specified in section 6.3.4.1.1 of [BCAST12-DVBSH-IPDC-Adaptation])</li> <li>13 – WiMAX Base Station Identifier (BSID) as defined in [IEEE 802.16-2004] and [IEEE 802.16e-2005]</li> <li>14 – WiMAX Operator ID (NAP ID) as defined in in [IEEE 802.16-2004] and [IEEE 802.16e-2005]</li> <li>15 – Forward Link Only Cell ID (specified in section 6.3 of [BCAST12-FLO-Adaptation])</li> <li>16 – DVB-SH DVB service ID (specified in section 6.3.4.1.2 of [BCAST12-DVBSH-IPDC-Adaptation]).</li> <li>17 – DVB-NGH Cell ID (specified in section 6.3.4.1. of [BCAST12-DVBNGH-Adaptation]</li> <li>18 – DVB-T2 Cell ID(specified in section 6.3.4.1. of [BCAST12-DVBNGH-Adaptation]</li> <li>19-127 reserved for future use</li> <li>128-255 reserved for proprietary use</li> </ul>	unsignedByte
------	---	-----------	---	---	--------------

CellArea	E4	NM/ TM	1..N	<p>The BDS specific transmit area given in the format as defined by type.</p> <p>Given that terminal’s current location is defined by one CellArea value (e.g. the current Cell ID) or multiple CellArea values (e.g. the transmitted DVB service IDs of the received DVB-SH Regionalized TS), multiple CellArea elements in same CellTargetArea element SHALL be interpreted as follows by the terminal, depending on polarity attribute of enclosing BroadcastArea element:</p> <ul style="list-style-type: none"> <li>when polarity is “true”, the terminal SHALL conclude it is in the area where service is intended for reception, if any of its current CellArea values is found in the list of Cell Area values of CellTargetArea.</li> <li>when polarity is “false”, the terminal SHALL conclude it is in the area where service is not intended for reception, if all its current CellArea values are found in the list of Cell Area values of CellTargetArea.</li> </ul> <p>Contains the following attribute: value</p> <p>Contains the following element: PP2CellID</p>	
value	A	NM/TM	1	The value of the cell ID. The structure of this value depends on the value of the type attribute	string
PP2CellID	E5	NO/ TO	0..N	<p>If type = 6, the value is Sector_ID as defined in [3GPP2 C.S0024-B]</p> <p>If type = 7, 8, 9 or 10, the value is BASE ID as defined in [3GPP2 C. S0002-E]</p> <p>3GPP2 terminals SHALL support this element.</p>	positiveInteger
lev_conf	E2	NO/TM	0..1	<p>The target level of confidence that the terminal is indeed located within the indicated ‘TargetArea’ as defined in [OMA MLP], used in performing the service reception filtering in accordance to polarity.</p> <p>Valid values: 0..100</p> <p>Note that lev_conf is allowed but less useful when target area corresponds to any of the allowed types of CellTargetArea, since it is presumed that air interface technology specific signalling informs the terminal whether or not it is currently located in the vicinity of the specified CellTargetArea”.</p>	unsignedByte

LocationFilter	E2	NO/TO	0..1	<p>Overall rule for use by the BCAST Terminal to make a location-based filtering decision regarding reception of the service referenced by this Service fragment. Besides targeted location(s), the rule MAY contain associated temporal and probabilistic presence/absence criteria.</p> <p>Contains the following elements:  LocationFilter1  LogicalOperation  LocationRequirement2  LocationFilter2</p> <p>Note: 'LocationRequirement2' and 'LocationFilter2' are mutually exclusive (choice) and SHALL appear once. If 'LogicalOperation' is a boolean operator (AND, OR) then LocationFilter1 SHALL appear.</p>	
LocationFilter1	E3		0..1	A constituent location filter in the formation of the overall location filter.	complexType as defined by E2 parent element 'LocationFilter'
LogicalOperation	E3	NO/TM	0..1	<p>If present, indicates the use of a logical binary or unary operation among constituent location criteria (location requirement or location filter), as operands, in producing the overall location filter. The following values are defined:</p> <ul style="list-style-type: none"> <li>0 – unspecified</li> <li>1 – Binary operator “AND”</li> <li>2 – Binary operator “OR”</li> <li>3 – Unary operator “NOT”</li> <li>4-127 – reserved for future use</li> <li>128-255 – reserved for proprietary use</li> </ul> <p>Note that binary operator “AND” and binary operator “OR” MAY be combined with unary operator “NOT” through iteration. Other operator combinations are not specified.</p>	unsignedByte

LocationRequirement2	E3	NO/TM	0..1	<p>Criterion for use by the BCAST Terminal to make a location-based filtering decision on reception of the service referenced by this Service fragment. Besides targeted location(s), this requirement MAY contain associated temporal and probabilistic presence/absence rules.</p> <p>Contains the following elements:  TargetArea  StartTime  EndTime  Duration  Lev_Conf_Present  Lev_Conf_Absent</p> <p>Note that if 'StartTime', 'EndTime' and 'Duration' are all absent, then the 'Lev_Conf_Present' or 'Lev_Conf_Absent' SHOULD be evaluated for the E4 'TargetArea' at the current time. In that case, the E2 'TargetArea' SHOULD be ignored if present.</p>	
TargetArea	E4	NO/TM	1..N	A target area filter.	complexType as defined by E2 child element 'TargetArea' of 'BroadcastArea'
StartTime	E4	NO/TM	0..1	In conjunction with 'EndTime', this element defines the target time window component of the location requirement conveyed by 'LocationRequirement2'. In the absence of 'Duration', 'LocationRequirement2' SHALL be evaluated over the entire duration of this target time window. This field contains the 32-bit integer part of an NTP time stamp.	unsignedInt
EndTime	E4	NO/TM	0..1	In conjunction with 'StartTime', this element defines the target time window component of the location requirement conveyed by 'LocationRequirement2'. In the absence of 'Duration', 'LocationRequirement2' SHALL be evaluated over the entire duration of this target time window. This field contains the 32-bit integer part of an NTP time stamp.	unsignedInt
Duration	E4	NO/TM	0..1	This element, if present, specifies the duration within the time interval defined by the difference between 'StartTime' and 'EndTime' for which 'LocationRequirement2' SHALL be evaluated. The unit of 'Duration' is seconds.	unsignedInt



Lev_Conf_Present	E4	NO/TM	0..1	<p>This element defines the minimum required level of confidence of the terminal being present in 'TargetArea' during the time window specified by 'StartTime' and 'EndTime' for at least an interval specified by 'Duration' in performing the content reception filtering in accordance to polarity. The value of this element SHALL be between 0 and 1 (inclusive).</p> <p>If 'LocationRequirement2' contains multiple target locations and is associated with a time window, 'Lev_Conf_Present' refers to the terminal being present in any of those locations during that time window for at least the interval specified by 'Duration'.</p>	decimal
Lev_Conf_Absent	E4	NO/TM	0..1	<p>This element defines the minimum required level of confidence of the terminal being absent in 'TargetArea' during the time window specified by 'StartTime' and 'EndTime' for at least an interval specified by 'Duration' in performing the content reception filtering in accordance to polarity. The value of this element SHALL be between 0 and 1 (inclusive).</p> <p>If 'LocationRequirement2' contains multiple target locations and is associated with a time window, 'Lev_Conf_Absent' refers to the terminal being absent in all of those locations during that time window for at least the interval specified by 'Duration'.</p>	decimal
LocationFilter2	E3	NO/TM	0..1	A constituent location filter associated with the logical operation defined above in producing the overall location filter.	complexType as defined by E2 parent element 'LocationFilter'

TermsOfUse	E1	NO/ TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>Contains the following attributes:</p> <p>type id userConsentRequired</p> <p>Contains the following elements:</p> <p>Country Language PreviewDataIDRef TermsOfUseText</p>	
type	A	NM/ TM	1	<p>The way the terminal SHALL interpret the Terms of Use:</p> <p>0 – Not used. 1 – Display before payout.</p> <p>If 'TermsOfUse' element of type '1' is present, terminal SHALL present the Terms of Use prior to playing out content or service associated with this fragment.</p> <p>2-127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
id	A	NM/ TM	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsent Required	A	NM/ TM	1	<p>Signals whether user consent for these Terms of Use is needed.</p> <p>true: User consent is required for these Terms of Use and needs to be confirmed. . How such confirmation is done is out of scope of this specification.</p> <p>false: User consent is not required for the Terms of Use.</p>	boolean

Country	E2	NM/ TM	0..N	List of countries for which the Terms of Use are applicable if consuming the service in that country. Each value is a Mobile Country Code according [ITU-MCC]. If this element is omitted, the Terms of Use are applicable to any country.	string of 3 digits
Language	E2	NM/ TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/ TM	0..1	Reference to the 'PreviewData' fragment which carries the representation of Terms of Use. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/ TM	0..1	Terms of Use text to be rendered. If this element is not present the 'PreviewDataIDRef' this element SHALL be present (Implementation in XML schema using <choice>).	string
Popularity	E1	NO/TO	0..1	Popularity of the associated service, measured by aggregate external opinion score, number of views of service content, or number of discussions generated by this service. This element may be used by the user in deciding whether or not to subscribe to the service. Contains the following attributes: rating noOfViews noOfDiscussions samplingDate	
rating	A	NM/TM	0..1	Rating of this service in terms of a user-generated value. This rating is normalized so as to fall in the range [0..1], with 0 being the lowest rating, and 1 being the highest possible rating.	decimal
noOfViews	A	NM/TM	0..1	Popularity of this service gathered from external review, expressed as the number of views content in this service.	unsignedInt
noOfDiscussions	A	NM/TM	0..1	Popularity of this content item in terms of the number of generated discussions and expressed as the number of comments or responses.	unsignedInt
samplingDate	A	NM/TM	0..1	The date and time when the Popularity statistics were sampled for transmission to the BCAST system. This attribute answers the question, "how old are the Popularity statistics?" This is expressed in UTC, using the 'dateTime' XML built-in datatype. If this attribute is present all Content fragments in this service (without samplingDates of their own) may be assumed to have been sampled on the date given by this attribute.	dateTime
PrivateExt	E1	NO/ TO	0..1	An element serving as a container for proprietary or application-specific extensions.	

<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	
------------------------	----	-------	------	---	--

### 5.1.2.2 Schedule

The ‘Schedule’ fragment specifies the time when content item(s) of a service are made available for distribution and/or presentation. For scheduling purposes a set of content items can be associated with each other through ‘Schedule’ fragment. In such cases the detailed breakdown of scheduling information can be hidden from the user. The Schedule fragment also provides means for providing choices of consumption. These are defined in detail in sections 5.8. In addition, the following rules apply:

- If there are two or more ‘Schedule’ fragments referring to Content or ‘Service’ fragments overlapping, i.e., valid at the same \*time\*, then one of these ‘Schedule’ fragments SHALL be declared to be the default.
- The terminal A/V player SHALL rely on timing information declared by the Session Description information independent of what is announced in the ‘Schedule’ fragment.

Examples for the use of the relations between ‘Service’, ‘Content’ and ‘Schedule’ fragment are given in Appendix C.

The ‘Schedule’ fragment also can specify the distribution schedule and the automatic activation time of the InteractivityMediaDocuments associated within the ‘InteractivityData’ fragment which is referenced within this ‘Schedule’.

Name	Type	Category	Cardinality	Description	Data Type
Schedule	E			‘Schedule’ fragment Contains the following attributes: id version defaultSchedule onDemand validFrom validTo UDBAllowed Contains the following elements: ServiceReference InteractivityDataReference ContentReference PreviewDataReference TermsOfUse PrivateExt	
id	A	NM/ TM	1	ID of the ‘Schedule’ fragment. The value of this attribute SHALL be globally unique.”	anyURI
version	A	NM/ TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no ‘validFrom’ attribute is given.	unsignedInt
defaultSchedule	A	NM/ TM	0..1	Defines whether this schedule is the default schedule of the content or service item it relates, in case there are multiple schedules valid at the same time, see also section 5.8. If present, this field SHALL only be set to true.	boolean

onDemand	A	NM/TM	0..1	<p>This attribute indicates the scheduled transmission of content that is accessible over the unicast channel where the transmission is not necessarily bounded into any specific interval of time. This type of content typically represents the "on-demand" component of a service.</p> <p>If this attribute is present and "true" it indicates that the referenced content item SHALL NOT be accessed automatically through this 'Schedule' fragment by the terminal when the associated service is selected and this 'Schedule' fragment is valid. This attribute SHALL NOT be present when this schedule fragment has no ContentReference element instantiated.</p> <p>Furthermore, this attribute in the 'Schedule' fragment MAY be set to 'true' if and only if the 'Access' fragments associated to the 'Schedule' fragment do not have the 'BroadcastServiceDelivery' element instantiated. See also section 5.8</p> <p>If this attribute is present and "false" it indicates that the referenced content item MAY be accessed automatically through this 'Schedule' fragment by the terminal when the associated service is selected and this 'Schedule' fragment is valid, see section 5.8.</p> <p>The default value of this attribute is "false".</p>	boolean
validFrom	A	NM/TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp	unsignedInt
validTo	A	NM/TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
UDBAllowed	A	NO TO	0..1	Represents whether if this Schedule can be used in User Defined Bundle subscriptions.	boolean
ServiceReference	E1	NM/TM	1	<p>Reference to the 'Service' fragment to which the 'Schedule' fragment belongs.</p> <p>If neither 'InteractivityDataReference' element nor 'ContentReference' element is present in the 'Schedule' fragment, then this 'Schedule' fragment defines the timeframe of the service availability as unbounded.</p> <p>Contains the following attributes:</p> <p>idRef</p>	
idRef	A	NM/TM	1	Identification of the 'Service' fragment which this 'Schedule' fragment is associated with.	anyURI

InteractivityDataReference	E1	NM/ TM	0..N	<p>Identification of the 'InteractivityData' fragment to which the 'Schedule' fragment is associated. Each occurrence of InteractivityDataReference implies the existence of a dedicated delivery schedule for those InteractivityMediaDocuments. This 'Schedule' fragment declares the available schedule for the file delivery of the InteractivityMediaDocuments referenced within the 'InteractivityData' fragment.</p> <p>Contains the following attributes: idRef</p> <p>Contains the following elements: AutoStart DistributionWindow</p> <p>Note: the presentation window is actually declared by the 'validFrom' and 'validTo' values in the InteractivityMediaDocument (see [BCAST12-Services] section 5.3.6).</p>	
idRef	A	NM/ TM	1	Identification of the 'InteractivityData' fragment which the 'Schedule' fragment relates to.	anyURI
AutoStart	E2	NM/ TM	0..N	<p>In the case the content or service the 'InteractivityData' fragment is related to is rendered, this element indicates the moment when the associated InteractivityMediaDocument is automatically activated if it is not rendered at that moment.</p> <p>If this element is not present or the terminal is turned on after 'AutoStart' time arrives, the associated InteractivityMediaDocument will not be automatically activated, but can be activated at any time of the validity of the media object document upon the user's request.</p> <p>It is RECOMMENDED that the terminal settings allow the users to configure whether to allow InteractivityMediaDocument to be automatically activated without users' request.</p> <p>This field contains the 32bits integer part of an NTP time stamp.</p>	unsignedInt
DistributionWindow	E2	NO/ TO	0..N	<p>Time interval in which the referenced InteractivityMediaDocument referenced within the 'InteractivityData' fragment is available for delivery.</p> <p>Contains the following attributes: startTime endTime duration id</p>	
startTime	A	NO/ TM	0..1	<p>Start of DistributionWindow. If not given, the validity is assumed to have begun at some time in the past.</p> <p>This field contains the 32bits integer part of an NTP time stamp.</p>	unsignedInt

endTime	A	NO/ TM	0..1	End of DistributionWindow. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
duration	A	NO/ TM	0..1	The maximum amount of time that the terminal which begins to acquire the Content item at the distribution endTime should allow to complete the acquisition. The unit of time is seconds.	unsignedInt
id	A	NO/ TO	0..1	Identification of the DistributionWindow. DistributionWindow id should be unique in the 'Schedule' fragment where it is declared.	unsignedInt
ContentReference	E1	NM/ TM	0..N	Reference to the 'Content' fragments that the 'Schedule' fragment relates to. Note: 'DistributionWindow' and/or 'PresentationWindow' can have multiple instances only for the Cachecast contents. For the other type of contents, 'PresentationWindow' and 'DistributionWindow' SHALL have zero or one instance. So if the same content is rebroadcasted, multiple 'ContentReference' element will be instantiated as multiple content fragments will be defined in accordance of its schedule. Contains the following attributes: idRef contentLocation repeatPlayback Contains the following elements: AutoStart DistributionWindow PresentationWindow	
idRef	A	NM/ TM	1	Identification of the 'Content' fragment which the 'Schedule' fragment relates to	anyURI

contentLocation	A	NM/ TM	0..1	<p>Identifies the location of the associated content item in the file or stream delivery session described by the 'Access' fragment during the schedule defined by this fragment.</p> <p>When ALC is used for file delivery, this corresponds to the 'Content-Location' attribute in the File element in the 'Access' fragment.</p> <p>When FLUTE is used for file delivery, this corresponds to the 'Content-Location' attribute in the FDT of the FLUTE session.</p> <p>When HTTP is used for file delivery, this attribute MAY be used to signal the 'abs_path' and 'query' components of the HTTP URL [RFC 2616] to address the requested file See section 5.5.2 in [BCAST12-Distribution] for information how the HTTP URL is constructed from this attribute and the 'AccessServerURL' element in the 'Access' fragment.</p> <p>When RTSP is used for stream delivery, this attribute MAY be used to signal the 'abs_path' component of the 'Request-URI' [RFC 2326] to be used in the request line of the RTSP request. See section 6.5 in [BCAST12-Distribution] for information how the RTSP Request-URI is constructed from this attribute and the 'AccessServerURL' element in the 'Access' fragment.</p> <p>When the stream delivery is made available over broadcast access and no control protocol is necessary, this attribute SHALL NOT be used.</p>	anyURI
repeatPlayback	A	NO/ TO	0..1	<p>Indicates whether the content item referenced by the Presentation Window and/or Distribution Window in the 'Schedule' fragment is of the repeat playback type.</p> <p>Default: false</p>	boolean
AutoStart	E2	NM/ TM	0..N	<p>Indicates the moment when the associated content is automatically activated if it is not rendered at that moment.</p> <p>If this element is not present, the associated content will not be automatically activated, but can be activated at any time upon the user's request.</p> <p>It is RECOMMENDED that the terminal settings allow the users to configure whether to allow content to be automatically activated without users' request.</p> <p>This field contains the 32bits integer part of an NTP time stamp.</p>	unsignedInt



Distribution Window	E2	NO/TO	0..N	Time interval in which the referenced content specified by ContentID is available for delivery. Contains the following attributes: startTime endTime duration id The terminal SHALL support this element if it supports Cachecast and Cachecast related services	
startTime	A	NO/TM	0..1	Start of DistributionWindow. If not given, the validity is assumed to have begun at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
endTime	A	NO/TM	0..1	End of DistributionWindow. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
duration	A	NO/TM	0..1	The maximum amount of time that the terminal which begins to acquire the Content item at the DistributionWindow 'endTime' should allow to complete the acquisition. The unit of time is in seconds	unsignedInt
id	A	NO/TO	0..1	Identification of the DistributionWindow which the each DistributionWindow relates to. DistributionWindow id should be unique in the 'Schedule' fragment where it is declared.	unsignedInt
Presentation Window	E2	NM/TM	0..N	Time interval in which the referenced content specified by ContentID is available for rendering. Contains the following attributes: startTime endTime duration id	
startTime	A	NM/TM	0..1	Start of PresentationWindow. If not given the validity is assumed is assumed to have begun at some time in the past. Note: When the content item is associated with the Cachecast service type, the startTime informs the user of the earliest instant he/she can begin the rendering of the associated content item. It does not represent the default start time at which the content is rendered by the terminal. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt

endTime	A	NM/ TM	0..1	End of PresentationWindow. If not given, the validity is assumed to end in undefined time in the future. Note: When the content item is associated with the Cachecast service type, the endTime informs the user of the latest instant he/she can begin the rendering of the associated content item. It does not represent the default end time at which the content ceases to be rendered by the terminal. Rendering SHOULD run to completion. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
duration	A	NM/ TM	0..1	Time duration of the referenced content for rendering. It informs the user the latest startTime for which the content item can be rendered in its entirety. This attribute is REQUIRED when the content item is associated with the Cachecast service type.	unsignedInt
id	A	NM/ TM	0..1	Identification of the PresentationWindow PresentationWindow id should be unique in the 'Schedule' fragment where it is declared.	unsignedInt
PreviewData Reference	E1	NM/ TM	0..N	Reference to the 'PreviewData' fragment which specifies the preview data (Eg. picture, video clip, or low-bit rate stream) associated with this schedule. The preview data associated with 'Schedule' fragment takes precedence over the preview data associated with the 'Content' fragments referenced by this 'Schedule' fragment. It is possible that there are more than one PreviewDataReference instances associated with the same fragment, in which case, the values of "usage" attributes of these PreviewDataReference instances SHALL be different. Contains the following attributes: idRef usage	
idRef	A	NM/ TM	1	Identification of the 'PreviewData' fragment which this fragment is associated with.	anyURI
usage	A	NM/ TM	1	Specifies the usage of the associated preview data. Possible values: 0. unspecified 1. Service-by-Service Switching 2. Service Guide Browsing 3. Service Preview 4. Barker 5. Alternative to blackout 6-127. reserved for future use 128-255. reserved for proprietary use The explanation and limitation on the above preview data usages is specified in section 5.7.	unsignedByte

TermsOfUse	E1	NO/ TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>Contains the following attributes:</p> <p>type id userConsentRequired</p> <p>Contains the following elements:</p> <p>Country Language PreviewDataIDRef TermsOfUseText</p>	
type	A	NM/ TM	1	<p>The way the terminal SHALL interpret the Terms of Use:</p> <p>0 – Not used.</p> <p>1 – Display before payout.</p> <p>If 'TermsOfUse' element of type '1' is present, terminal SHALL present the Terms of Use prior to playing out content or service associated with this fragment</p> <p>2-127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
id	A	NM/ TM	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsent Required	A	NM/ TM	1	<p>Signals whether user consent for these Terms of Use is needed.</p> <p>true: User consent is required for these Terms of Use and needs to be confirmed. How such confirmation is done is out of scope of this specification.</p> <p>false: User consent is not required for the Terms of Use.</p>	boolean
Country	E2	NM/ TM	0..N	<p>List of countries for which the Terms of Use are applicable if consuming the scheduled content in that country. Each value is a Mobile Country Code according [ITU-MCC].</p> <p>If this element is omitted, the Terms of Use are applicable to any country.</p>	string of 3 digits

Language	E2	NM/ TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/ TM	0..1	Reference to the 'PreviewData' fragment which carries the representation of Terms of Use. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/ TM	0..1	Terms of Use text to be rendered. If this element is not present, the 'PreviewDataIDRef' element SHALL be present (Implementation in XML schema using <choice>).	string
PrivateExt	E1	NO/ TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.1.2.3 Content

'Content' fragment describes a content or programme that is a part of a service that the content refers to. A 'Content' fragment can be composed of several (audiovisual) media streams originating from several sources which together form a logical entity. An example is a single TV show broadcasted in several audio languages. A 'Content' fragment contains the metadata to describe a content or programme, represented by the attributes and elements in the following table. In the case of 'ParentalRating', 'TargetUserProfile', 'Genre', and 'BroadcastArea', should one or more of these elements exist in both the 'Content' fragment and its associated 'Service' fragment, the values of those elements in the 'Content' fragment take precedence over its counterparts in 'Service' fragment.

Name	Type	Category	Cardinality	Description	Data Type
------	------	----------	-------------	-------------	-----------

Content	E			<p>‘Content’ fragment</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>id</li> <li>version</li> <li>validFrom</li> <li>validTo</li> <li>globalContentID</li> <li>emergency</li> <li>baseCID</li> <li>UDBAllowed</li> <li>amAllowed</li> </ul> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>ServiceReference</li> <li>ProtectionKeyID</li> <li>Name</li> <li>Description</li> <li>StartTime</li> <li>EndTime</li> <li>AudioLanguage</li> <li>TextLanguage</li> <li>Length</li> <li>ParentalRating</li> <li>TargetUserProfile</li> <li>Genre</li> <li>Extension</li> <li>PreviewDataReference</li> <li>BroadcastArea</li> <li>TermsOfUse</li> <li>Popularity</li> <li>Freshness</li> <li>Terminal Provisioning</li> <li>PrivateExt</li> </ul>	
amAllowed	A	NO/TO	0..1	<p>This attribute indicates whether Audience Measurement using the Audience Measurement solutions specified in section 5.20 of [BCAST12-Services] is allowed or not for this content. If the value of amAllowed is true or the attribute is not present in the ‘Content’ fragment then the measurement is allowed. If the value of amAllowed is false, Measurement is not allowed.</p> <p>Note 1: The explicitly signalled True or False value of amAllowed in the ‘Content’ fragment overrides the value of amAllowed in the ‘Service’ fragment for this content.</p> <p>Note 2: Usage of amAllowed is further described in section 5.20 of [BCAST12-Services]</p>	Boolean
id	A	NM/ TM	1	ID of the ‘Content’ fragment. The value of this attribute SHALL be globally unique.	anyURI

version	A	NM/ TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the 'validFrom' attribute, or as soon as it has been received if no 'validFrom' attribute is given.	unsignedInt
validFrom	A	NM/ TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
globalContentID	A	NM/ TM	0..1	The globally unique identifier identifying the content that this 'Content' fragment describes. If the content item identified by this attribute belongs to the 'Auxiliary Data' service type, it is expected that 'globalContentID' will have a matching value in the 'GlobalContentID' sub-element of an 'AuxDataTrigger' notification message whose <type> = 0 (i.e. download trigger) as specified in (Section 5.14.3 of [BCAST12-Services]).	anyURI
emergency	A	NO/ TO	0..1	When assigned with value 'true', specifies that this content is content of emergency nature. This attribute can be used for presentation purposes to users. It is RECOMMENDED that the Terminal processes the reception of the services or content of emergency nature with high priority, and highlights their availability to user. How to order the emergency service or content is out of the scope of the specification. The default value of this attribute is 'false'.	boolean

baseCID	A	NO/ TO	0..1	<p>For the DRM Profile, part of the Service or Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. The Service or Program CID is obtained from the BaseCID as described in [BCAST12-ServContProt], section 5.5.1].</p> <p>In case this element is present the terminal SHALL use this element to identify the corresponding asset within an OMA DRM 2.0 Rights Object, instead of the baseCID(s) of the ‘Service’ fragment(s) this ‘Content’ fragment is associated with.</p> <p>In case this ‘Content’ fragment contains a reference to a ‘Service’ fragment this element MAY be present when:</p> <ul style="list-style-type: none"> <li>– this ‘Content’ fragment is referenced by a ‘PurchaseItem’ fragment or when</li> <li>– a ‘PurchaseItem’ fragment references a ‘Schedule’ fragment that references this ‘Content’ fragment,</li> </ul> <p>to identify the corresponding asset within an OMA DRM 2.0 Rights Object, in case the network supports the DRM profile.</p> <p>In case this ‘Content’ fragment does not contain a reference to a ‘Service’ fragment this element SHALL be present when:</p> <ul style="list-style-type: none"> <li>– this ‘Content’ fragment is referenced by a ‘PurchaseItem’ fragment or when</li> <li>– a ‘PurchaseItem’ fragment references a ‘Schedule’ fragment that references this ‘Content’ fragment</li> </ul> <p>to identify the corresponding asset within an OMA DRM 2.0 Rights Object, in case the network supports the DRM profile.</p> <p>The network and terminal SHALL support this element in case the DRM Profile is supported [BCAST12-ServContProt].</p> <p>Note: for uniqueness of the baseCID see Appendix H.</p>	string
UDBAllowed	A	NO TO	0..1	Represents whether if this Service can be used in User Defined Bundle subscriptions/	boolean

ServiceReference	E1	NM/ TM	0..N	Reference to the 'Service' fragment(s) to which the 'Content' fragment belongs. Contains the following attributes: idRef weight Note: If the content item described by this 'Content' fragment belongs to a service of the 'Auxiliary Data' service type, then this content item SHOULD NOT be described in the Program Guide. Specifically, for 'Auxiliary Data' services, those elements and attributes in the Program Guide portion of this fragment that allow a minimum cardinality of 0 SHALL not be instantiated	
idRef	A	NM/ TM	1	Identification of the 'Service' fragment which this 'Content' fragment is associated with.	anyURI
weight	A	NM/ TM	0..1	Intended order of display of this 'Content' fragment relative to other 'Content' fragments belonging to the same service as presented to the end user. The order of display is by increasing Weight value (i.e., content with lowest Weight is displayed first). Default: 65535	unsignedShort
ProtectionKeyID	E1	NO/ TO	0..N	Key identifier needed to access protected content. This information allows the terminal to determine whether or not it has the correct key material to access content item(s) within a PurchaseItem. In a scenario where this fragment is shared among multiple service providers, different key identifiers from the different service providers to access this specific protected content item may be mixed in this element and the terminal SHOULD be able to sort out the key identifiers associated with the terminal's affiliated service provider based on the Key Domain ID. How this is used is out of scope and is left to implementation. The network and terminal SHALL support this element in case the Smartcard Profile is supported [BCAST12-ServContProt]. The protection key identifiers to access a specific service or content item SHALL only be signalled in one of the following fragment types: 'Service', 'Content', 'PurchaseItem'; 'PurchaseData' or 'Access' fragments, but not mixed. Contains the following attributes: type min max	base64Binary



type	A	NM/TM	1	<p>Type of ProtectionKeyID:</p> <p>0: ProtectionKeyID is the 5-byte long concatenation of the Key Domain ID with the Key group part of the SEK/PEK ID, where both values are as used in the Smartcard Profile [BCAST12-ServContProt].</p> <p>The Key number part SHALL NOT be provided.</p> <p>The Terminal MAY use the Key Domain ID and Key group part of the ProtectionKeyID to determine whether it already has either the SEK applicable to access the service to which this content item belongs, or the PEK applicable to this content item. The Terminal MAY use this information to indicate to the user which content items can currently be accessed. The Terminal SHALL not use the SEK/PEK ID in the ProtectionKeyID to request a missing SEK or PEK. It is possible for the Terminal to request missing SEK or PEK based on the information from the secure function after the STKM decryption has been failed.</p> <p>1-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte
min	A	NM/TM	0..1	<p>Indicates the lower validity value of the key needed to access the service / content.</p> <p>For type 0, corresponds to the value of the lowest timestamp (32 bits) of an STKM needed to access the service / content, as used in the Smartcard Profile [BCAST12-ServContProt]</p>	nonNegativeInteger
max	A	NM/TM	0..1	<p>Indicates the higher validity value of the key needed to access the service / content.</p> <p>For type 0, corresponds to the value of the highest timestamp (32 bits) of an STKM needed to access the service / content, as used in the Smartcard Profile [BCAST12-ServContProt].</p>	nonNegativeInteger

				<p style="text-align: center;"><b>Start of program guide</b></p> <p>The program guide elements of this fragment are grouped between the Start of program guide and end of program guide cells in this fragment.</p> <p>The program guide elements are for user interpretation. This enables the content creator to provide user readable information about the service. The terminal SHOULD use all declared program guide elements in this fragment for presentation to the end-user. The terminal MAY offer search, sort etc functionalities.</p> <p>The Program Guide consists of the following elements:</p> <p>Name Description StartTime EndTime AudioLanguage TextLanguage Length ParentalRating TargetUserProfile Genre Extension</p> <p>Note: The elements 'Popularity' and 'Freshness' MAY also be included in the Program Guide for display to the end user.</p>	
Name	E1	NM/ TM	1..N	Name of the 'Content' fragment, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
Description	E1	NM/ TM	0..N	Description, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
StartTime	E1	NM/ TM	0..1	The start time of the content which is for presentation purposes to the end user , expressed in UTC, using 'dateTime' XML built-in datatype. This element is applicable for scheduled rendering of non-Cachecast content. For Cachecast content, the start time is defined by the 'startTime' attribute of the 'PresentationWindow' element in the 'Schedule' fragment.	dateTime
EndTime	E1	NM/ TM	0..1	The end time of the content which is for presentation purposes to the end user, expressed in UTC, using 'dateTime' XML built-in datatype. This element is applicable for scheduled rendering of non-Cachecast content. For Cachecast content, the end time is defined by the 'endTime' attribute of the 'PresentationWindow' element in the 'Schedule' fragment.	dateTime

AudioLanguage	E1	NM/ TM	0..N	<p>This element declares for the end users that this content is available with an audio stream corresponding to the language represented by the value of this element.</p> <p>The textual value of this element can be made available for the end users in different languages. In such a case the language used to represent the value of this element is signalled using the built-in XML attribute 'xml:lang'. See section 7 Multi-language support.</p> <p>Contains the following attribute: languageSDPtag</p>	string
languageSDPtag	A	NM/ TO	1	<p>Identifier of the audio language described by the parent 'AudioLanguage' element as used in the media sections describing the audio track in a Session Description.</p> <ul style="list-style-type: none"> <li>The 'languageSDPtag' SHALL be formatted according to the rules of [RFC 3066], for the described language.</li> <li>Each 'AudioLanguage' element declaring the same audio stream SHALL have the same value of the 'languageSDPtag'.</li> </ul>	string
TextLanguage	E1	NM/ TM	0..N	<p>This element declares for the end user that the textual components of this content are available in the language represented by the value of this element. The textual components can be, for instance, a caption or a sub-title track.</p> <p>The textual value of this element can be made available for the end users in different languages. In such a case the language used to represent the value of this element is signalled using the built-in XML attribute 'xml:lang'. See section 7 Multi-language support.</p> <p>The same rules and constraints as specified for the element 'AudioLanguage' of assigning and interpreting the attributes 'languageSDPtag' and 'xml:lang' SHALL be applied for this element also.</p> <p>Contains the following attribute: languageSDPtag</p>	string
languageSDPtag	A	NM/TO	1	<p>Identifier of the text language described by the parent 'TextLanguage' element as used in the media sections describing the textual track in a Session Description.</p>	string
Length	E1	NM/ TM	0..1	Duration of the A/V content declared	duration

ParentalRating	E1	NM/ TM	0..N	<p>The ParentalRating element defines criteria parents may use to determine whether the associated item is suitable for access by children, defined according to the regulatory requirements of the service area.</p> <p>The parental rating level defined for ‘Content’ fragment overrides the rating level defined for the corresponding ‘Service’ fragment during the validity of the ‘Schedule’ fragment.</p> <p>If there are multiple content items associated with a ‘Schedule’ fragment with different parental ratings, then the one with the most restrictive parental rating overrides the others.</p> <p>The terminal SHALL support ‘ParentalRating’ being a free string, and the terminal MAY support the structured way to express the parental rating level by using the ‘ratingSystem’ and ‘ratingValueName’ attributes as defined below.</p> <p>Contains the following attributes: ratingSystem ratingValueName</p>	string
ratingSystem	A	NO/ TM	0..1	<p>Specifies the parental rating system in use, in which context the value of the ‘ParentalRating’ element is semantically defined. This allows terminals to identify the rating system in use in a non-ambiguous manner and act appropriately.</p> <p>This attribute SHALL be instantiated when a rating system is used. Absence of this attribute means that no rating system is used (i.e. the value of the ‘ParentalRating’ element is to be interpreted as a free string).</p> <p>If this attribute is instantiated:</p> <ul style="list-style-type: none"> <li>– The value of this attribute SHALL be one of the ‘rating_type’ values as listed in the OMA BCAST Parental Rating System Registry at [OMNA].</li> <li>– The ‘ParentalRating’ element SHALL contain the string representation of a number that is a valid ‘rating_value’ in this particular rating system.</li> <li>– This attribute MAY contain the value ‘10’ (OMA BCAST generic rating scheme), allowing to define a rating value in a non-registered parental rating system. In such case, the ‘ParentalRating’ element SHALL contain the string representation of a number between 1 and 255, 1 being the least and 255 being the most restrictive rating value. As these values are generic, the human-readable label of that rating value SHALL be signalled in the attribute ‘ratingValueName’.</li> </ul>	unsignedByte

ratingValue Name	A	NO/TM	0..1	The human-readable name of the rating value given by this ParentalRating element. This attribute SHALL be present in case the 'ratingSystem' attribute contains the value '10'.	string
TargetUserP rofile	E1	NO/ TO	0..N	Profile attributes of the users whom the content is targeting at. The detailed personal attribute names and the corresponding values are specified by attributes of 'attributeName' and 'attributeValue'. Amongst the possible profile attribute names are age, gender, occupation, etc. (subject to national/local rules & regulations, if present and as applicable regarding use of personal profiling information and personal data privacy). The extensible list of 'attributeName' and 'attributeValue' pairs for a particular content enables end user profile filtering and end user preference filtering of broadcast contents. The terminal SHOULD be able to support 'TargetUserProfile' element. The terminal behavior for interpreting and acting upon 'TargetUserProfile' is out of the scope. It is RECOMMENDED that use of 'TargetUserProfile' element is an "opt-in" capability for users. Terminal settings SHOULD allow users to configure whether to input their personal profile or preference and whether to allow broadcast content to be automatically filtered based on the users' personal attributes without users' request. Contains the following attributes: attributeName attributeValue	
attributeNam e	A	NM/ TM	1	Profile attribute name.	string
attributeValu e	A	NM/ TM	1	Profile attribute value.	string

Genre	E1	NM/ TM	0..N	<p>Classification of content associated with characteristic form (e.g. comedy, drama). The OMA BCAST Service Guide allows describing the format of the Genre element in the Service Guide in two ways:</p> <ul style="list-style-type: none"> <li>• The first way is to use a free string</li> <li>• The second way is to use the “href” attributes of the Genre element to convey the information in the form of a controlled vocabulary (classification scheme as defined in [TVA-Metadata] or classification list as defined in [MIGFG]).</li> </ul> <p>The built-in XML attribute xml:lang MAY be used with this element to express the language. The Network MAY instantiate several different sets of ‘Genre’ element, using it as a free string or with a ‘href’ attribute. The Network SHALL ensure the different sets have equivalent and non-conflicting meaning, and the terminal SHALL select one of the sets to interpret for the end-user. Contains the following attributes: type href</p>	string
type	A	NO/ TO	0..1	<p>This attribute signals the level of this ‘Genre’ element. The following values are allowed: “main” “secondary” “other”</p>	string

href	A	NO/ TO	0..1	<p>This attribute signals the controlled vocabulary used for this ‘Genre’ element.</p> <p>If this attribute is supported, the following applies to the support and use of classification schemes according to [TVA-Metadata]:</p> <ul style="list-style-type: none"> <li>for values of the ‘type’ attribute equal to "main" or "secondary", the terminal MAY support levels 1-4 of the TV Anytime ContentCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:ContentCS:2005 as defined in Annex A.8 of [TVA-Metadata]</li> <li>for a value of the ‘type’ attribute equal to "other", the terminal MAY support levels 1-3 of the TV Anytime IntendedAudienceCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:IntendedAudienceCS:2005 as defined in Annex A.11 of [TVA-Metadata]. When the IntendedAudienceCS is provided simultaneously with an instantiation of the ‘TargetUserProfile’, the two SHALL have equivalent meaning.</li> <li>The network SHALL use the following URI syntax to signal terms from classification schemes: &lt;classificationSchemeURI&gt; “:” &lt;termID&gt;</li> <li>If this attribute is instantiated by the network, the element ‘Genre’ SHALL be an empty string and the xml:lang attribute SHALL NOT be used.</li> </ul> <p>If this attribute is supported, the following applies to the support and use of the classification from [MIGFG]:</p> <ul style="list-style-type: none"> <li>This classification SHALL be signalled with the URI “http://www.loc.gov/rr/mopic/miggen.html”</li> <li>The string value carried in the ‘Genre’ element SHALL be used to convey the actual value of the classification as given in [MIGFG]</li> <li>The Network MAY use the values “main” and “secondary” of the ‘type’ attribute so as to provide an ordering of two classification applying to the same Service.</li> </ul> <p>Other Classification Schemes MAY be signalled with the 'href' attribute, however how they are used is out of scope of this specification.</p> <p>If this attribute is not instantiated, the ‘Genre’ element SHALL be a free string.</p>	anyURI
------	---	-----------	------	--	--------

Extension	E1	NM/ TM	0..N	Additional information related to this fragment. Contains the following attribute: url Contains the following element: Description	
url	A	NM/ TM	1	URL containing additional information related to this fragment.	anyURI
Description	E2	NM/TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
<b>End of program guide</b>					
PreviewData Reference	E1	NM/ TM	0..N	Reference to the 'PreviewData' fragment which specifies the preview data (Eg. picture, video clip, or low-bit rate stream) associated with this content. It is possible that there are more than one PreviewDataReference instances associated with the same fragment, in which case, the values of "usage" attributes of these PreviewDataReference instances SHALL be different. Contains the following attributes: idRef usage	
idRef	A	NM/ TM	1	Identification of the 'PreviewData' fragment which this fragment is associated with.	anyURI
usage	A	NM/ TM	1	Specifies the usage of the associated preview data. Possible values: 0. unspecified 1. Service-by-Service Switching 2. Service Guide Browsing 3. Service Preview 4. Barker 5. Alternative to blackout 6-127. reserved for future use 128-255. reserved for proprietary use The explanation and limitation on the above preview data usages is specified in section 5.7.	unsignedByte



BroadcastArea	E1	NO/TO	0..1	<p>Broadcast area to include location information for BCAST contents</p> <p>Contains the following attribute:</p> <p>Polarity</p> <p>filteringTime</p> <p>Contains the following elements:</p> <p>TargetArea</p> <p>lev_conf</p> <p>LocationFilter</p> <p>BCAST 1.1 terminals that are location-capable SHALL use only LocationFilter, polarity and filteringTime if present. Otherwise if the LocationFilter element is not present, BCAST 1.1 terminals that are location-capable SHALL use polarity, filteringTime TargetArea and lev_conf to compute location presence.</p>	
polarity	A	NO/TO	0..1	<p>Indication of whether the associated target area is intended for positive or negative terminal reception of the content item.</p> <p>If polarity = true or 1, this indicates the associated content item is intended for reception by terminals located within the corresponding geographical area. (Default)</p> <p>If polarity = false or 0, this indicates the associated content item is not intended for reception by terminals located within the corresponding geographical area.</p>	boolean
filteringTime	A	NO/TO	0..1	<p>For cachecast service/content, this attribute indicates whether the filtering against target area occurs at the service-guide fragment broadcast time, content download time, or the content rendering time. The terminal SHOULD obtain its current location information before performing filtering.</p> <p>0 – the download time specified by the DistributionWindow in the corresponding Schedule fragment.</p> <p>1 – the rendering time specified by the PresentationWindow in the corresponding Schedule fragment.</p> <p>2 – the location-based filter is applied as soon as this fragment is received by the terminal.</p> <p>3-255 – reserved for future use.</p> <p>Note that when this attribute is present, BCAST 1.1 terminals SHOULD perform filtering at the time indicated by the "filteringTime" value whereas BCAST 1.0 terminals ignore this attribute. The service provider should create their content filters with this limitation in mind.</p>	unsignedByte

TargetArea	E2	NO/ TM	0..N	The target area to distribute contents (as specified in the [OMA MLP] with modifications) Contains the following elements: shape cc mcc name_area ZipCode CellTargetArea Only one of the above six elements SHALL be instantiated at the same time. Implementation in XML schema using <choice>.	
shape	E3	NO/ TM	0..1	Shapes used to represent a geographic area that describes (as specified in the [OMA MLP])	
cc	E3	NO/ TM	0..1	Country code as specified in [OMA MLP] , e.g. 355 for Albania	unsignedShort
mcc	E3	NO/ TM	0..1	Mobile country code, 3 digits e.g. 276 for Albania (as specified in [ITU-MCC], aligned with [OMA MLP])	string of three digits
name_area	E3	NO/ TM	0..N	Geopolitical name of area such as 'Seoul' (as specified in the [OMA MLP]). The instances of 'name_area' element differ only in language. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
ZipCode	E3	NO/ TM	0..1	Zip code	string
CellTargetArea	E3	NO/ TM	0..1	The target area to distribute content specified by the BDS specific service coverage area or minimum transmit area Contains the following attribute: type Contains the following element: CellArea	

type	A	NM/ TM	1	<p>Allowed values are:</p> <p>0 – Unspecified</p> <p>1 – 3GPP Cell Global Identifier as defined in [3GPP TS 23.003]</p> <p>2 – 3GPP Routing Area Identifier (RAI) as defined in [3GPP TS 23.003]</p> <p>3 – 3GPP Location Area Identifier (LAI) as defined in [3GPP TS 23.003]</p> <p>4 – 3GPP Service Area Identifier (SAI) as defined in [3GPP TS 23.003]</p> <p>5 – 3GPP MBMS Service Area Identity (MBMS SAI) as defined in [3GPP TS 23.003]</p> <p>6 – 3GPP2 Subnet ID as defined in [3GPP2 X.S0022-A]</p> <p>7 – 3GPP2 SID as defined in [3GPP2 C.S0005-E]</p> <p>8 – 3GPP2 SID+NID as defined in [3GPP2 C.S0005-E]</p> <p>9 – 3GPP2 SID+NID+PZID as defined in [3GPP2 C.S0005-E]</p> <p>10 – 3GPP2 SID+PZID as defined in [3GPP2 C.S0005-E]</p> <p>11 – DVB-H Cell ID (specified in section 6.3.4.1 of [BCAST12-DVBH-IPDC-Adaptation] )</p> <p>12 – DVB-SH Cell ID (specified in section 6.3.4.1.1 of [BCAST12-DVBSH-IPDC-Adaptation])</p> <p>13 – WiMAX Base Station Identifier (BSID) as defined in [IEEE 802.16-2004] and [IEEE 802.16e-2005]</p> <p>14 – WiMAX Operator ID (NAP ID) as defined in in [IEEE 802.16-2004] and [IEEE 802.16e-2005]</p> <p>15 – Forward Link Only Cell ID (specified in section 6.3 of [BCAST-11-FLO-Adaptation])</p> <p>16 – DVB-SH DVB service ID (specified in section 6.3.4.1.2 of [BCAST12-DVBSH-IPDC-Adaptation]).</p> <p>17 – DVB-NGH Cell ID(specified in section 6.3.4.1. of [BCAST12-DVBNGH-Adaptation]</p> <p>18 – DVB-T2 Cell ID(specified in section 6.3.4.1. of [BCAST12-DVBNGH-Adaptation]</p> <p>19-127 reserved for future use</p> <p>128-255 reserved for proprietary use</p>	<b>unsignedByte</b>
------	---	-----------	---	--	---------------------

CellArea	E4	NM/ TM	1..N	<p>The BDS specific transmit area given in the format as defined by type.</p> <p>Given that terminal's current location is defined by one CellArea value (e.g. the current Cell ID) or multiple CellArea values (e.g. the transmitted DVB service IDs of the received DVB-SH Regionalized TS), multiple CellArea elements in same CellTargetArea element SHALL be interpreted as follows by the terminal, depending on polarity attribute of enclosing BroadcastArea element:</p> <ul style="list-style-type: none"> <li>when polarity is "true", the terminal SHALL conclude it is in the area where content item is intended for reception, if any of its current CellArea values is found in the list of Cell Area values of CellTargetArea.</li> <li>when polarity is "false", the terminal SHALL conclude it is in the area where service is not intended for reception, if all its current CellArea values are found in the list of Cell Area values of CellTargetArea.</li> </ul> <p>Contains the following attribute: Value</p> <p>Contains the following element: PP2CellID</p>	
value	A	NM/TM	1	The value of the cell ID. The structure of this value depends on the value of the type attribute.	string
PP2CellID	E5	NO/ TO	0..N	<p>If type = 6, the value is Sector_ID as defined in [3GPP2 C.S0024-B]</p> <p>If type = 7, 8, 9 or 10, the value is BASE ID as defined in [3GPP2 C. S0002-E]</p> <p>Note: See relevant BDS specification for the data type of this element</p> <p>Note: 3GPP2 terminals SHALL support this element</p>	positiveInteger
lev_conf	E2	NO/TM	0..1	<p>The target-level of confidence that the terminal is indeed located within the indicated 'TargetArea' as defined in [OMA MLP], used in performing the content reception filtering in accordance to polarity.</p> <p>Valid values: 0..100</p> <p>Note that lev_conf is allowed but less useful when target area corresponds to any of the allowed types of CellTargetArea, since it is presumed that air interface technology specific signalling informs the terminal whether or not it is currently located in the vicinity of the specified CellTargetArea".</p>	unsignedByte

LocationFilter	E2	NO/TO	0..1	<p>Overall rule for use by the BCAST Terminal to make a location-based filtering decision regarding reception of the service referenced by this Service fragment. Besides targeted location(s), the rule MAY contain associated temporal and probabilistic presence/absence criteria.</p> <p>Contains the following elements:</p> <p>LocationFilter1 LogicalOperation LocationRequirement2 LocationFilter2</p> <p>Note: 'LocationRequirement2' and 'LocationFilter2' are mutually exclusive (choice) and SHALL appear once. If 'LogicalOperation' is a boolean operator (AND, OR) then LocationFilter1 SHALL appear.</p>	
LocationFilter1	E3		0..1	A constituent location filter in the formation of the overall location filter.	complexType as defined by E2 parent element 'LocationFilter'
LogicalOperation	E3	NO/TM	0..1	<p>If present, indicates the use of a logical binary or unary operation among constituent location criteria (location requirement or location filter), as operands, in producing the overall location filter. The following values are defined:</p> <p>0 – unspecified 1 – Binary operator “AND” 2 – Binary operator “OR” 3 – Unary operator “NOT” 4-127 – reserved for future use 128-255 – reserved for proprietary use</p> <p>Note that binary operator “AND” and binary operator “OR” MAY be combined with unary operator “NOT” through iteration. Other operator combinations are not specified.</p>	unsignedByte

LocationRequirement2	E3	NO/TM	0..1	<p>Criterion for use by the BCAST Terminal to make a location-based filtering decision on reception of the service referenced by this Service fragment. Besides targeted location(s), this requirement MAY contain associated temporal and probabilistic presence/absence rules.</p> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>TargetArea</li> <li>StartTime</li> <li>EndTime</li> <li>Duration</li> <li>Lev_Conf_Present</li> <li>Lev_Conf_Absent</li> </ul> <p>Note that if 'StartTime', 'EndTime' and 'Duration' are all absent, then the 'Lev_Conf_Present' or 'Lev_Conf_Absent' SHOULD be evaluated for the E4 'TargetArea' at the current time. In that case, the E2 'TargetArea' SHOULD be ignored if present.</p>	
TargetArea	E4	NO/TM	1..N	A target area filter.	complexType as defined by E2 child element 'TargetArea' of 'BroadcastArea'
StartTime	E4	NO/TM	0..1	In conjunction with 'EndTime', this element defines the target time window component of the location requirement conveyed by 'LocationRequirement2'. In the absence of 'Duration', 'LocationRequirement2' SHALL be evaluated over the entire duration of this target time window. This field contains the 32-bit integer part of an NTP time stamp.	unsignedInt
EndTime	E4	NO/TM	0..1	In conjunction with 'StartTime', this element defines the target time window component of the location requirement conveyed by 'LocationRequirement2'. In the absence of 'Duration', 'LocationRequirement2' SHALL be evaluated over the entire duration of this target time window. This field contains the 32-bit integer part of an NTP time stamp.	unsignedInt
Duration	E4	NO/TM	0..1	This element, if present, specifies the duration within the time interval defined by the difference between 'StartTime' and 'EndTime' for which 'LocationRequirement2' SHALL be evaluated. The unit of 'Duration' is seconds.	unsignedInt

Lev_Conf_Present	E4	NO/TM	0..1	<p>This element defines the minimum required level of confidence of the terminal being present in 'TargetArea' during the time window specified by 'StartTime' and 'EndTime' for at least an interval specified by 'Duration' in performing the content reception filtering in accordance to polarity. The value of this element SHALL be between 0 and 1 (inclusive).</p> <p>If 'LocationRequirement2' contains multiple target locations and is associated with a time window, 'Lev_Conf_Present' refers to the terminal being present in any of those locations during that time window for at least the interval specified by 'Duration'.</p>	decimal
Lev_Conf_Absent	E4	NO/TM	0..1	<p>This element defines the minimum required level of confidence of the terminal being absent in 'TargetArea' during the time window specified by 'StartTime' and 'EndTime' for at least an interval specified by 'Duration' in performing the content reception filtering in accordance to polarity. The value of this element SHALL be between 0 and 1 (inclusive).</p> <p>If 'LocationRequirement2' contains multiple target locations and is associated with a time window, 'Lev_Conf_Absent' refers to the terminal being absent in all of those locations during that time window for at least the interval specified by 'Duration'.</p>	decimal
LocationFilter2	E3	NO/TM	0..1	A constituent location filter associated with the logical operation defined above in producing the overall location filter.	complexType as defined for E2 parent element 'LocationFilter'

TermsOfUse	E1	NO/ TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>Contains the following attributes:</p> <p>type id userConsentRequired</p> <p>Contains the following elements:</p> <p>Country Language PreviewDataIDRef TermsOfUseText</p>	
type	A	NM/ TM	1	<p>The way the terminal SHALL interpret the Terms of Use:</p> <p>0 – Not used.</p> <p>1 – Display before payout.</p> <p>If 'TermsOfUse' element of type '1' is present, terminal SHALL present the Terms of Use prior to playing out content or service associated with this fragment.</p> <p>2-127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
id	A	NM/ TM	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsent Required	A	NM/ TM	1	<p>Signals whether user consent for these Terms of Use is needed.</p> <p>true: User consent is required for these Terms of Use and needs to be confirmed. How such confirmation is done is out of scope of this specification.</p> <p>false: User consent is not required for the Terms of Use.</p>	boolean
Country	E2	NM/ TM	0..N	<p>List of countries for which the Terms of Use are applicable if consuming the content in that country. Each value is a Mobile Country Code according [ITU-MCC].</p> <p>If this element is omitted, the Terms of Use are applicable to any country.</p>	string of 3 digits



Language	E2	NM/ TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/ TM	0..1	Reference to the 'PreviewData' fragment which carries the representation of Terms of Use. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/ TM	0..1	Terms of Use text to be rendered. If this element is not present, the 'PreviewDataIDRef' element SHALL be present (Implementation in XML schema using <choice>).	string
Popularity	E1	NO/TO	0..1	Popularity of the associated content item, measured by aggregate external opinion score, number of views of the content item, or number of discussions generated by this item. This element may be used by the terminal in deciding whether or not to download and cache the content item referenced by this 'Content' fragment, for subsequent user accessibility. Such filtering MAY be performed in conjunction with stored user preference or user profile information. Contains the following attributes: Rating noOfViews noOfDiscussions samplingDate	
rating	A	NM/TM	0..1	Rating of this content item in terms of a user-generated value. This rating is normalized so as to fall in the range [0..1], with 0 being the lowest rating, and 1 being the highest possible rating.	decimal
noOfViews	A	NM/TM	0..1	Popularity of this content item gathered from external review, expressed as the number of views of this content.	unsignedInt
noOfDiscussions	A	NM/TM	0..1	Popularity of this content item in terms of the number of generated discussions and expressed as the number of comments or responses.	unsignedInt
samplingDate	A	NM/TM	0..1	The date and time when the Popularity statistics were sampled for transmission to the BCAST system. This attribute answers the question, "how old are the Popularity statistics?" This is expressed in UTC, using the 'dateTime' XML built-in datatype. This attribute, if present, overrides the samplingDate in all Service fragments associated with this content.	dateTime

Freshness	E1	NO/TO	0..1	Temporal freshness of the associated content item and its attributes. This element may be use by the terminal in deciding whether or not to download and cache the content item referenced by this 'Content' fragment, for subsequent user accessibility. Such filtering MAY be performed in conjunction with stored user preference or user profile information. Contains the following attributes: releaseDate broadcastDate	
releaseDate	A	NM/TM	0..1	The earliest release date of the content (i.e. when this content first appeared in Theaters, on DVD, or on television or elsewhere). This is expressed in UTC, using the 'dateTime' XML built-in datatype.	dateTime
broadcastDate	A	NM/TM	0..1	The earliest broadcast date (i.e. when this content first appeared on broadcast media, such as television or a BCAST system.) This is expressed in UTC, using the 'dateTime' XML built-in datatype.	dateTime
TerminalProvisioning	E1	NO/TO	0..1	Broadcast Terminal Provisioning specific information. This element is used to provide filtering information so that only the targeted terminals retrieve the Terminal Provisioning Package. It is RECOMMENDED that a Terminal Provisioning Service be instantiated and the DependencyReference element in the related PurchaseData fragment be instantiated so that automatic subscription to the Terminal Provisioning Service is enabled allowing Terminal Provisioning Content fragments to be checked by terminals. This feature is based upon OMA-DM [DM 1.3]. For details on using Broadcast Terminal Provisioning refer to TS Services [BCAST12-Services]. This feature is not related to smartcard provisioning and only related to the Terminal Consists of the following attribute: type Consists of the following element: Target Note: The detailed scope of Broadcast Terminal Provisioning is to be defined by DM WG.	
type	A	NM/TM	1	The type of Terminal Provisioning. Allowed values are: 0 – Firmware update [FUMO] 1 – Software update [SCOMO] 2 – Device Capability Control 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte

Target	E2	NM/TM	1..N	Filtering information for targeted terminals. Contains the following attributes: version manufacturer model hardware dm Contains the following element: Extensions	
version	A	NM/TM	0..1	Version of the announced update. MAY be used for filtering in Firmware.	unsignedInt
manufacturer	A	NM/TM	0..1	Targeted manufacturer name of the terminal. MAY be used for filtering in Firmware, Software and Device Capability Control.	string
model	A	NM/TM	0..1	Targeted model name/number of the terminal. MAY be used for filtering in Firmware, Software and Device Capability Control.	string
hardware	A	NM/TM	0..1	Version of the hardware of the targeted device. May be used for filtering in Firmware	string
dm	A	NM/TM	0..1	Version of the OMA DM enabler required for the Terminal Provisioning service. Allowed values are: 0 – DM 1.3 1 – DM 2.0 2 – FUMO 1.0 3 – SCoMO 1.0 4 – DiagMon 1.0 5-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
extensions	E3	NM/TM	0..N	Supplementary extensions filters Consists of the following attributes: attributeName attributeValue	
attributeName	A	NM/NM	1	Extension filter attribute name	string
attributeValue	A	NM/TM	1	Extension filter attribute value	string
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

#### 5.1.2.4 Access

An ‘Access’ fragment describes to the terminal how it can access a service or a schedule during the lifespan of the ‘Access’ fragment. If the service or content is protected, the fragment also contains service and content protection information.

Name	Type	Category	Cardinality	Description	Data Type
------	------	----------	-------------	-------------	-----------

Access	E			<p>‘Access’ fragment</p> <p>Contains the following attributes:</p> <p>id version validFrom validTo</p> <p>Contains the following elements:</p> <p>AccessType KeyManagementSystem EncryptionType ServiceReference ScheduleReference TerminalCapabilityRequirement BandwidthRequirement ServiceClass ReferredSGInfo PreviewDataReference NotificationReception SmartcardProvisioningReception PrivateExt</p>	
id	A	NM/ TM	1	ID of the ‘Access’ fragment. The value of this attribute SHALL be globally unique.	anyURI
version	A	NM/ TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the validFrom attribute, or as soon as it has been received if no validFrom attribute is given.	unsignedInt
validFrom	A	NM/ TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
AccessType	E1	NM/ TM	1	<p>Defines the type of access.</p> <p>Note: Either one of ‘BroadcastServiceDelivery’ or ‘UnicastServiceDelivery’ but not both SHALL be instantiated. Implementation in XML Schema should use &lt;choice&gt;.</p> <p>Contains the following elements:</p> <p>BroadcastServiceDelivery UnicastServiceDelivery</p>	
BroadcastServiceDelivery	E2	NM/ TM	0..1	<p>This element is used for the indication of IP transmission.</p> <p>Contains the following elements:</p> <p>BDSType SessionDescription FileDescription</p>	

BDSType	E3	NM/ TM	0..1	Identifier of the type of underlying distribution system that this 'Access' fragment relates to. Contains the following element: Type Version	
Type	E4	NM/ TM	0..1	Type of underlying BDS, possible values: 0. IPDC over DVB-H 1. 3GPP MBMS 2. 3GPP2 BCMCS 3. IPDC over DVB-SH 4. WiMAX 5. Forward Link Only 6. DVB-NGH 7. DVB-T2 8-127. reserved for future use 128-255. reserved for proprietary use	unsignedByte
Version	E4	NM/ TM	0..N	Version of underlying BDS. Possible values for MBMS are specified according to the following syntax in ABNF [RFC4234]:  <pre> version = "3GPP." release "." bearer release = "R6" / "R7" / "R8"  bearer = "GERAN" / "UTRAN" / "MBSFN-FDD" / "MBSFN-TDD" / "MBSFN-IMB" </pre> By applying these rules, strings such as 3GPP.R6.UTRAN and 3GPP.R8.MBSFN-IMB can be constructed. To allow backwards compatibility with BCAST 1.0, the strings "Rel-6" and "Rel-7" (which do not follow the ABNF syntax above) are also included in the set of possible values for MBMS.  Other possible values include, e.g., 1x or HRPD or Enhanced HRPD for BCMCS.	string

SessionDescription	E3	NM/TM	1	<p>Reference to or inline copy of a Session Description information associated with this 'Access' fragment that the media application in the terminal uses to access the service.</p> <p>Note: a referenced 'SessionDescription' fragment may be delivered in two ways: via broadcast or via fetch over interaction channel.</p> <p>In the case of fetch over interaction channel, the 'SessionDescription' fragment can be acquired by accessing the URI (given as attribute of the different Session Description reference elements).</p> <p>Contains the following elements:</p> <p>SDP SDPRef USBDRef ADPRef</p> <p>The presence of elements 'SDP', 'SDPRef', and USBDRef are mutually exclusive. This is enforced in the XML schema by using &lt;choice&gt;.</p>	
SDP	E4	NM/TM	0..1	<p>An inlined Session Description in SDP format [RFC 4566] that SHALL either be embedded in a CDATA section or base64-encoded.</p> <p>Contains the following attribute:</p> <p>encoding</p>	string
encoding	A	NM/TM	0..1	<p>This attribute signals the way the Session Description has been embedded:</p> <ul style="list-style-type: none"> <li>• It SHALL NOT be present when the Session Description is embedded into a CDATA section.</li> <li>• It SHALL be present and set to "base64" in case the Session Description is base64-encoded.</li> </ul>	string
SDPRef	E4	NM/TM	0..1	<p>Reference to a Session Description in SDP format [RFC 4566]</p> <p>Contains the following attributes:</p> <p>uri idRef</p> <p>If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.</p>	
uri	A	NM/TM	0..1	<p>The URI referencing an external resource containing SDP information. This URI is used for interactive retrieval.</p>	anyURI
idRef	A	NM/TM	0..1	<p>The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique</p>	anyURI

USBDRef	E4	NM/TM	0..1	Reference to an instance of MBMS User Service Bundle Description as specified in [3GPP TS 26.346] section 5.2.2, with the restrictions defined in section 5.1.2.5 of this spec. Contains the following attributes: uri idRef If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.	
uri	A	NM/TM	0..1	The URI referencing an external resource containing MBMS-USBD information. This URI is used for interactive retrieval.	anyURI
idRef	A	NM/TM	0..1	The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique	anyURI
ADPRef	E4	NM/TM	0..1	Reference to an AssociatedDeliveryProcedure for File and Stream Distribution as specified in [BCAST12-Distribution] section 5.3.4. Contains the following attributes: uri idRef If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.	
uri	A	NM/TM	0..1	The URI referencing an external resource containing AssociatedDeliveryProcedure for File and Stream Distribution. This URI is used for interactive retrieval.	anyURI
idRef	A	NM/TM	0..1	The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique	anyURI
FileDescription	E3	NO/TM	0..1	File metadata for file delivery sessions. This element SHALL be provided when ALC is used. This element SHALL NOT be used in conjunction with FLUTE. The network SHALL support 'FileDescription' element and all its sub-elements and attributes if ALC is used for File Distribution function. Contains the following attributes: Content-Type Content-Encoding FEC-OTI-FEC-Encoding-ID FEC-OTI-FEC-Instance-ID FEC-OTI-Maximum-Source-Block-Length FEC-OTI-Encoding-Symbol-Length FEC-OTI-Max-Number-of-Encoding-Symbols FEC-OTI-Scheme-Specific-Info Contains the following elements: File	
Content-Type	A	NO/TM	0..1	See RFC 3926, section 3.4.2	string

Content-Encoding	A	NO/TM	0..1	See RFC 3926, section 3.4.2	string
FEC-OTI-FEC-Encoding-ID	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedByte
FEC-OTI-FEC-Instance-ID	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Maximum-Source-Block-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Encoding-Symbol-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Max-Number-of-Encoding-Symbols	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Scheme-Specific-Info	A	NO/TM	0..1	This attribute MAY be used to communicate FEC information which is not adequately represented by the other attributes related to FEC.	base64Binary
File	E4	NM/TM	1..N	Parameters of a file. Contains the following attributes: Content-Location TOI Content-Length Transfer-Length Content-Type Content-Encoding Content-MD5 FEC-OTI-FEC-Encoding-ID FEC-OTI-FEC-Instance-ID FEC-OTI-Maximum-Source-Block-Length FEC-OTI-Encoding-Symbol-Length FEC-OTI-Max-Number-of-Encoding-Symbols FEC-OTI-Scheme-Specific-Info	
Content-Location	A	NM/TM	1	See RFC 3926, section 3.4.2	anyURI
TOI	A	NM/TM	1	See RFC 3926, section 3.4.2	positiveInteger
Content-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
Transfer-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
Content-Type	A	NO/TM	0..1	See RFC 3926, section 3.4.2	string
Content-Encoding	A	NO/TM	0..1	See RFC 3926, section 3.4.2	string



Content-MD5	A	NO/TM	0..1	See RFC 3926, section 3.4.2	base64Binary
FEC-OTI-FEC-Encoding-ID	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedByte
FEC-OTI-FEC-Instance-ID	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Maximum-Source-Block-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Encoding-Symbol-Length	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Max-Number-of-Encoding-Symbols	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedLong
FEC-OTI-Scheme-Specific-Info	A	NO/TM	0..1	This attribute MAY be used to communicate FEC information which is not adequately represented by the other attributes related to FEC.	base64Binary
UnicastServiceDelivery	E2	NM/TM	0..N	This element indicates which server and/or protocol is used for delivery of service over Interaction Channel. Contains the following attribute: type Contains the following elements: AccessServerURL SessionDescription	
type	A	NM/TM	1	Specifies transport mechanism that is used for this access. 0 – HTTP 1 – WAP 1.0 2 – WAP 2.x 3 – Generic RTSP to initialize RTP delivery 4 – RTSP to initialize RTP delivery as per 3GPP-PSS (3GPP packet-switched streaming service) 5 – RTSP to initialize RTP delivery as per 3GPP2-MSS (3GPP2 multimedia streaming services) 6 – FLUTE over Unicast 7-127 Reserved for future use 128-255 Reserved for proprietary use Note: Specification or negotiation of ports used for unicast service delivery is handled by the used unicast distribution mechanisms. For example, RTSP and PSS based systems (values 3 and 4) do port negotiation within the RTSP signalling exchange.	unsignedByte

AccessServerURL	E3	NM/ TM	0..N	<p>Server URL which the terminal can use to construct the URL to receive the service via the Interaction Network as specified in section 5.5 and 6.5 of [BCAST12-Distribution]. If there are multiple instances of AccessServerURL signalled, the terminal SHALL randomly select one of them, and keep using it for the duration of the session.</p> <p>In case of HTTP, this element SHALL contain the fields 'http' and 'host' according to [RFC 2616] and MAY contain the fields 'port', 'abs_path' and 'query'. In case of RTSP, this element SHALL contain the fields 'rtsp' and 'host' according to [RFC 2326] and MAY contain the fields 'port' and 'abs_path'.</p> <p>Note that this information MAY be complemented or overridden by information signalled in the 'contentLocation' attribute in the 'Schedule' fragment, or the 'VideoURI', 'AudioURI', and 'PictureURI' elements in the 'PreviewData' fragment. See sections 5.5.2 and 6.5 in [BCAST12-Distribution] for information how the RTSP Request-URI and the HTTP URL are constructed from this element and said the 'contentLocation' attribute.</p> <p>If 'type' attribute has one of the values "3", "4" or "5" either E3 element 'SessionDescription' or E3 element 'AccessServerURL' or both SHALL be instantiated.</p>	anyURI
-----------------	----	-----------	------	--	--------

SessionDescription	E3	NM/ TM	0..1	<p>Reference to or inline copy of a Session Description information associated with this 'Access' fragment that the media application in the terminal uses to access the service.</p> <p>Note: a referenced 'SessionDescription' fragment may be delivered in two ways: via broadcast or via fetch over interaction channel.</p> <p>In the case of fetch over interaction channel, the 'SessionDescription' fragment can be acquired by accessing the URI (given as attribute of the different Session Description reference elements).</p> <p>Contains the following elements:</p> <p>SDP SDPRef USBDRef ADPRef</p> <p>The presence of elements 'SDP', 'SDPRef', and USBDRef are mutually exclusive. This is enforced in the XML schema by using &lt;choice&gt;.</p> <p>If 'SessionDescription' E3 element is instantiated, and the 'type' attribute has one of the values "3", "4" or "5", the terminal MAY use it to acquire Session Description information (including RTSP Control URL) via broadcast channel or interaction channel using 'SDPRef' or use inlined SDP (E4 element 'SDP'), instead of fetching Session Description information via RTSP. Further, in this case, 'AccessServerURL' E3 element MAY NOT be present.</p> <p>If 'type' attribute has one of the values "3", "4" or "5" either E3 element 'SessionDescription' or E3 element 'AccessServerURL' or both SHALL be instantiated.</p>	
SDP	E4	NM/ TM	0..1	<p>An inlined Session Description in SDP format [RFC 4566] that SHALL either be embedded in a CDATA section or base64-encoded.</p> <p>Contains the following attribute:</p> <p>encoding</p>	string
encoding	A	NM/TM	0..1	<p>This attribute signals the way the Session Description has been embedded:</p> <ul style="list-style-type: none"> <li>• It SHALL NOT be present when the Session Description is embedded into a CDATA section.</li> <li>• It SHALL be present and set to "base64" in case the Session Description is base64-encoded.</li> </ul>	string

SDPRef	E4	NM/ TM	0..1	Reference to a Session Description in SDP format [RFC 4566] Contains the following attributes: uri idRef If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.	
uri	A	NM/ TM	0..1	The URI referencing an external resource containing SDP information. This URI is used for interactive retrieval. The terminal SHALL support HTTP URI for this purpose.	anyURI
idRef	A	NM/ TM	0..1	The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique	anyURI
USBDRef	E4	NM/TM	0..1	Reference to an instance of MBMS User Service Bundle Description as specified in [3GPP TS 26.346] section 5.2.2, with the restrictions defined in section 5.1.2.5 of this spec. Contains the following attributes: uri idRef If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.	
uri	A	NM/ TM	0..1	The URI referencing an external resource containing MBMS-USBD information. This URI is used for interactive retrieval.	anyURI
idRef	A	NM/ TM	0..1	The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique	anyURI
ADPRef	E4	NM/TM	0..1	Reference to an AssociatedDeliveryProcedure for File and Stream Distribution as specified in [BCAST12-Distribution] section 5.3.4. Contains the following attributes: uri idRef If both 'uri' and 'idRef' are present, the referenced Session Description information SHALL be identical.	
uri	A	NM/ TM	0..1	The URI referencing an external resource containing AssociatedDeliveryProcedure for File and Stream Distribution. This URI is used for interactive retrieval.	anyURI
idRef	A	NM/ TM	0..1	The id of the referenced 'SessionDescription' fragment if the fragment is delivered over the broadcast channel in SGDU, globally unique	anyURI

KeyManagementSystem	E1	NM/ TM	0..N	<p>Information of Key Management System(s)(KMS) that can be used to contact the BCAST Permissions Issuer and, in case of the SmartCard Profile whereby GBA is used for SMK derivation, whether GBA_U is mandatory or whether either GBA_ME or GBA_U can be used.</p> <p>Note that the BCAST Permissions Issuer can support more than one KMS.</p> <p>If KeyManagementSystem is not instantiated, it means neither Service nor Content Protection is applied "to the access method declared by the parent fragment.</p> <p>Multiple occurrences of 'KeyManagementSystem' elements are allowed within this fragment only if all of the 'KeyManagementSystem' elements have different 'kmsType' attribute.</p> <p>Contains the following elements: ProtectionKeyID PermissionsIssuerURI TerminalBindingKeyID</p> <p>Contains the following attributes: kmsType secureChannelRequired protectionType</p>	
kmsType	A	NM/ TM	1	<p>Identifies the type of Key Management System(s)(KMS). Possible values:</p> <p>0. oma-bcast-drm-pki Indicates OMA BCAST DRM profile (Public Key Infrastructure)</p> <p>1. oma-bcast-gba_u-mbms Indicates BCAST Smartcard profile using GBA_U (Symmetric Key Infrastructure)</p> <p>2. oma-bcast-gba_me-mbms Indicates BCAST Smartcard profile using GBA_ME</p> <p>3. oma-bcast-prov-bcmcs Indicates provisioned 3GPP2 BCMCS SKI</p> <p>4-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte

secureChannelRequired	A	NO/TO	0..1	<p>This attribute signals whether the protection of the interface between Terminal and Smartcard using the Secure Channel (see section 6.13 of [BCAST12-ServContProt]) is required for the protection of the Service/Content delivered via this Access. Note that this signalling is indicative; the enforcement is done in the Smartcard as described in section 6.13 of [BCAST12-ServContProt].</p> <p>This attribute is only meaningful in the Smartcard Profile, i.e. it SHALL NOT be instantiated in case the kmsType attribute has a value of '0' or '2'.</p> <p>If Secure Channel support is required, this attribute SHALL be instantiated and set to 'true'. It SHALL be omitted or set to 'false' otherwise.</p> <p>Terminals supporting the Smartcard Profile SHALL support this attribute. For these Terminals, the following applies:</p> <ul style="list-style-type: none"> <li>Terminals MAY use this attribute to indicate whether the Service/Content delivered via this Access can be consumed, depending on whether or not the Secure Channel is supported by the combination of Terminal and Smartcard.</li> <li>Further, Terminals that do not support the Secure Channel in combination with the Smartcard SHALL NOT offer to the user the option to purchase any PurchaseItem that includes Services or Content which are only accessible through Accesses requiring the Secure Channel.</li> </ul>	boolean
protectionType	A	NM/TM	1	<p>Specifies the protection type offered by the KMS. Values:</p> <p>0. Content protection only, as defined by the LTKM (protection_after_reception in STKM = 0x00 or 0x01 [BCAST12-ServContProt])</p> <p>1. Service protection only (protection_after_reception in STKM = 0x03 [BCAST12-ServContProt])</p> <p>2. Content protection as defined by LTKM, plus playback of protected recording permission (protection_after_reception in STKM = 0x02 [BCAST12-ServContProt])</p> <p>3-127 Reserved for future use</p> <p>128-255 Reserved for proprietary use</p> <p>This attribute may also be used for presentation purpose to users, to indicate whether the content item(s), associated with the referenced Schedule by this 'Access' fragment, is protected or not.</p>	unsignedByte

PermissionsIssuerURI	E2	NM/TM	1	<p>The semantics of this element are dependent on the type of the key management system.</p> <p>In case of the DRM Profile, the value of this element corresponds to the ‘RightsIssuerURL’ as defined by [DRMDRM-v2.0].</p> <p>In case of the Smartcard profile, this element represents the address where the terminal performs the following three service provisioning procedures:</p> <ul style="list-style-type: none"> <li>• Registration Procedure (section 5.1.6.7 in [BCAST12-Services]),</li> <li>• LTKM Request Procedure (Section 5.1.6.8 in [BCAST12-Services]) and</li> <li>• Deregistration Procedure (section 5.1.6.9 in [BCAST12-Services]).</li> </ul> <p>In case the terminal needs to issue service provisioning messages other than three listed above, the terminal SHALL send these requests to the address given by the ‘PurchaseURL’ element of the corresponding ‘PurchaseChannel’ fragment.</p> <p>For both the Smartcard and the DRM profile, if Adapted PDCF is used, the value of this element gives the value of the ‘RightsIssuerURL’ or ‘KeyIssuerURL’ as specified for the Adapted PDCF file format (section 8.3.2 in [BCAST12-ServContProt]).</p> <p>In the case of the Smartcard Profile, the hostname part of the URI is the Fully Qualified Domain Name of the BSM.</p> <p>In the case of (U)SIM Smartcard Profile, the path component of PermissionsIssuerURI SHALL be equal to ‘/keymanagement’, and the terminal SHALL append a ‘requesttype’ parameter to the URI, with a value set to ‘register’, ‘deregister’ or ‘msk-request’ for respectively registration, deregistration and LTKM request procedure. In this case, the resulting Request-URI SHALL be compliant with respectively sections G.2.1, G.2.2 and G.2.3 of [3GPP TS 33.246 v7]. As specified in [3GPP TS 33.246 v7], the terminal MAY add additional URI parameters to this Request-URI.</p>	anyURI
----------------------	----	-------	---	--	--------

ProtectionKeyID	E2	NO/TO	0..N	<p>Key identifier needed to access protected service/content. This information allows the terminal to determine whether or not it has the correct key material to access service(s)/content item(s) within a PurchaseItem. In a scenario where this fragment is shared among multiple service providers, different key identifiers from the different service providers to access this specific protected service/content item may be mixed in this element and the terminal SHOULD be able to sort out the key identifiers associated with the terminal's affiliated service provider based on the Key Domain ID. How this is used is out of scope and is left to implementation.</p> <p>The network and terminal SHALL support this element in case the Smartcard Profile is supported [BCAST12-ServContProt].</p> <p>The protection key identifiers to access a specific service or content item SHALL only be signalled in one of the following fragment types: 'Service', 'Content', 'PurchaseItem', 'PurchaseData' or 'Access' fragments, but not mixed.</p> <p>Contains the following attribute: type</p>	base64Binary
type	A	NM/TM	1	<p>Type of ProtectionKeyID:</p> <p>0: ProtectionKeyID is the 5-byte long concatenation of theKey Domain ID with the Key group part of the SEK/PEK ID, where both values are as used in the Smartcard Profile [BCAST12-ServContProt].</p> <p>The Key number part SHALL NOT be provided. The Terminal MAY use the Key Domain ID and Key group part of the ProtectionKeyID to determine whether it already has SEK/PEK applicable to access the related service/content item. The Terminal MAY use this information to indicate to the user which services/content items can currently be accessed. The Terminal SHALL not use the SEK/PEK ID in the ProtectionKeyID to request a missing SEK or PEK. It is possible for the Terminal to request missing SEK or PEK based on the information from the secure function after the STKM decryption has been failed.</p> <p>1-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte



TerminalBindingKeyID	E2	NO/ TO	0..1	<p>Number identifying the Terminal Binding Key ID (TBK ID) that is needed to access the service.</p> <p>If the element is absent, no TerminalBindingKey is used.</p> <p>Both the network and the terminal with the Smartcard Profile SHALL support this element. It is TM for terminals with the smartcard profile.</p> <p>This element does not apply to the DRM profile.</p> <p>Contains the following attribute: tbkPermissionsIssuerURI</p>	unsignedInt
tbkPermissionsIssuerURI	A	NO/ TM	0..1	<p>Specifies the Permissions Issuer URI for the TerminalBindingKey if it is different from the 'PermissionsIssuerURI' sub-element of the 'KeyManagementSystem' element.</p> <p>If the attribute is not present the same 'PermissionsIssuerURI' indicated for KeyManagementSystem is used to acquire both SEK / PEK and TerminalBindingKey.</p>	anyURI
Encryption Type	E1	NM/ TM	0..N	<p>Specifies which encryption methods the terminal is to be able to support in order to utilize this Access. Contains the same value as traffic_protection_protocol signalled in STKM.</p> <p>0 – IPsec 1 – SRTP 2 – ISMACryp 3 – DCF 4-255 – Reserved for future use.</p> <p>If this element is not present, this Access is not encrypted.</p>	unsignedByte
ServiceReference	E1	NM/ TM	0..N	<p>Reference to the 'Service' fragment(s) to which the 'Access' fragment belongs.</p> <p>Either one of 'ServiceReference' or 'ScheduleReference', or neither, but not both SHALL be instantiated.</p> <p>Each 'Service' fragment SHALL be associated to at least one 'Access' fragment to enable the terminal to access the Service.</p> <p>A single 'Access' fragment MAY reference to multiple 'Service' fragments. This is used when there are several independent descriptions of a single Service.</p>	
idRef	A	NM/ TM	1	<p>Identification of the 'Service' fragment which this 'Access' fragment is associated with.</p>	anyURI

ScheduleReference	E1	NM/ TM	0..N	<p>Reference to the ‘Schedule’ fragment(s) to which the ‘Access’ fragment belongs.</p> <p>This provides a reference to a ‘Schedule’ fragment to temporarily override the default ‘Access’ fragment of the Service addressed by the Schedule.</p> <p>Either one of ‘ServiceReference’ or ‘ScheduleReference’, or neither, but not both SHALL be instantiated. Note: Implementation in XML Schema using &lt;choice&gt;.</p> <p>Contains the following attribute: idRef</p> <p>Contains the following element: DistributionWindowID</p>	
idRef	A	NM/ TM	1	<p>Identification of the ‘Schedule’ fragment which the ‘Access’ fragment relates to.</p>	anyURI
DistributionWindowID	E2	NO/ TM	0..N	<p>Relation reference to the DistributionWindowID to which the ‘Access’ fragment belongs.</p> <p>The ‘DistributionWindowID’ element declared in this element SHALL be the complete collection or a subset of the DistributionWindow ids declared in the ‘Schedule’ fragment, to which ‘idRef’ reference belongs.</p>	unsignedInt
TerminalCapabilityRequirement	E1	NO/ TM	0..1	<p>Terminal capabilities needed to consume the service or content.</p> <p>This element provides a hint to the terminal of what is needed to apply the consumption method represented by this ‘Access’ fragment. It is out of scope of this specification how the terminal applies this information.</p> <p>For video and audio, the media type and the related media type attributes in the Session Description information signal the audio/video decoder. Additionally, this element provides a hint on the decoder requirements to the terminal that it can use without inspecting the Session Description information. For initiation of the media consumption session, the parameters defined in the Session Description information SHALL take priority.</p> <p>It is RECOMMENDED that the complexities of the audio/video streams are described here if they differ from the complexities which can be derived from the media type attributes in the Session Description information (e.g. level)</p> <p>Contains the following elements: Video Audio DownloadFile Rich Media</p>	

Video	E2	NO/ TM	0..1	Video codec capability related requirements Contains the following elements: MIMEType Complexity	
MIMEType	E3	NO/ TM	0..1	MIME Media type of the video. If the complexities that can be derived from the MIMEType element and the codec parameters below differ from the parameters defined under the 'Complexity' element below, then the parameters defined under the 'Complexity' element SHALL take priority. Contains the following attribute: codec	string
codec	A	NO/ TM	0..1	The codec parameters for the associated MIME Media type. If the MIME type definition specifies mandatory parameters, these MUST be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the media SHOULD be included in the string. One example of the parameters defined for video/3GPP, video/3GPP2 is specified in [RFC4281].	string
Complexity	E3	NM/ TM	0..1	The complexity the video decoder has to deal with. It is RECOMMENDED that this element is included if the complexity indicated by the MIME type and codec parameters differs from the actual complexity. Contains the following elements: Bitrate Resolution MinimumBufferSize	
Bitrate	E4	NO/ TM	0..1	The total bit-rate of the video stream. Contains the following attributes: average maximum	
average	A	NO/ TM	0..1	The average bit-rate in kbit/s	unsignedShort
maximum	A	NO/ TM	0..1	The maximum bit-rate in kbit/s	unsignedShort
Resolution	E4	NO/ TM	0..1	The resolution of the video. Contains the following attributes: horizontal vertical temporal	
horizontal	A	NM/ TM	1	The horizontal resolution of the video in pixels.	unsignedShort
vertical	A	NM/ TM	1	The vertical resolution of the video in pixels.	unsignedShort
temporal	A	NM/ TM	1	The maximum temporal resolution in frames per second.	decimal

MinimumBufferSize	E4	NO/TM	0..1	The minimum decoder buffer size needed to process the video content in kbytes.	unsignedInt
Audio	E2	NO/TM	0..1	The audio codec capability. Contains the following elements: MIMEType Complexity	
MIMEType	E3	NO/TM	0..1	MIME Media type of the audio. If the complexities that can be derived from the MIMEType element and the codec parameters below differ from the parameters defined under the 'Complexity' element below, then the parameters defined under the 'Complexity' element SHALL take priority. Contains the following attribute: codec	string
codec	A	NO/TM	0..1	The codec parameters for the associated MIME Media type. If the MIME type definition specifies mandatory parameters, these MUST be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the media SHOULD be included in the string. One example of the parameters defined for audio/3GPP, audio/3GPP2 is specified in [RFC4281].	string
Complexity	E3	NM/TM	0..1	The complexity the audio decoder has to deal with. It is RECOMMENDED that this element is included if the complexity indicated by the MIME type and codec parameters differs from the actual complexity. Contains the following elements: Bitrate MinimumBufferSize	
Bitrate	E4	NO/TM	0..1	The total bit-rate of the audio stream. Contains the following attributes: average maximum	
average	A	NO/TM	0..1	The average bit-rate in kbit/s	unsignedShort
maximum	A	NO/TM	0..1	The maximum bit-rate in kbit/s	unsignedShort
MinimumBufferSize	E4	NO/TM	0..1	The minimum decoder buffer size needed to process the audio content in kbytes.	unsignedInt
DownloadFile	E2	NO/TM	0..1	The required capability for the download files. Contains the following elements: MIMEType	

MIMETYPE	E3	NM/ TM	1..N	<p>Assuming a download service consists of a set of files with different MIME types which together make up the service, the terminal must support all of these MIME types in order to be able to present the service to the user.</p> <p>The format of this string</p> <ul style="list-style-type: none"> <li>• SHALL follow the ‘Content-Type’ syntax defined in [RFC 2045].</li> <li>• Additionally the ‘Content-Type’ MAY be augmented as defined in [RFC 4281].</li> </ul> <p>In the latter case the ‘Content-Type’ SHALL begin by</p> <ul style="list-style-type: none"> <li>• “audio/3gpp”,</li> <li>• “audio/3gpp2”,</li> <li>• “video/3gpp”,</li> <li>• “video/3gpp2”</li> </ul> <p>Contains the following attribute: codec</p>	string
codec	A	NO/ TM	0..1	<p>The codec parameters for the associated MIME Media type.</p> <p>If the file's MIME type definition specifies mandatory parameters, these MUST be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the file SHOULD be included in the string. One example of the parameters defined for audio/3GPP, audio/3GPP2, video/3GPP, video/3GPP2 is specified in [RFC4281].</p>	string
RichMedia	E2	NO/ TO	0..1	<p>Declares the rich media content characteristics. How to instantiate this element is further detailed in section 5.18 of [BCAST12-Services].</p> <p>Contains the following element: Capabilities</p>	
Capabilities	E3	NM/TM	1	<p>Contains the following attributes: type version</p> <p>Contains the following element: MIMETYPE Complexity</p>	
type	A	NM/TM	1	<p>Indicate the type of RM content.</p> <p>Allowed values are: 0 – according to MIME type 1 – W3C SVG Tiny 2 – OMA RME 3 – MPEG LAsE R 4 – 3GPP DIMS 5-127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
version	A	NM/TM	1	Defines the version associated with the type	string

MIMEType	E4	NM/ TM	0..N	MIME Media type of the rich media content. Contains the following attribute: Codec	String
codec	A	NM/ TM	0..1	The codec parameters for the associated MIME Media type. If the file's MIME type definition specifies mandatory parameters, these <b>MUST</b> be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the file <b>SHOULD</b> be included in the string.	string
Complexity	E4	NM/ TM	1	The complexity the rich media engine has to deal with. Contains the following attributes: profile sceneUpdateCommands screenOrientation Contains the following elements: Animations MediaElements DOMnodes Scripting Compression	
profile	A	NO/TO	1	Defines the profile/level of the RMS Note: it is conditional on the 'type' and 'version' attributes	string
sceneUpdate Commands	A	NO/ TM	1	Indicates whether the rich media content requires the processing of scene update commands to render the content. Default: False	boolean
screenOrient ation	A	NO/ TM	1	Indicates whether the rich media scene requires scene orientation management Default: False	boolean
Animations	E5	NO/ TM	1	The number of animations in the rich media scene. Contains the following attributes: maximum	
maximum	A	NM/ TM	0..1	The maximum number of animations in the rich media scene Default: 0	unsignedSh ort
MediaEleme nts	E5	NM/ TM	1	The number of media elements (i.e. video and audio) embedded in or referenced by the rich media scene. Contains the following attributes: simultaneousVideo simultaneousAudio	
simultaneous Video	A	NM/ TM	0..1	The maximum number of concurrently running video elements in the rich media scene. Default: 0	unsignedSh ort
simultaneous Audio	A	NM/ TM	0..1	The maximum number of concurrently running audio elements in the rich media scene. Default: 0	unsignedSh ort

DOMNodes	E5	NO/ TM	1	The number of DOM nodes required to render the rich media content This element SHALL be present if the Rich Media scene description is DOM-based, and SHALL be omitted otherwise. Contains the following attributes: maximum	
maximum	A	NM/ TM	1	The maximum number of active DOM nodes in a rich media scene at any given time.	unsignedShort
Scripting	E5	NO/ TM	1	Indicates whether the rich media scene requires the processing of scripts (for e.g. ECMAScript) to render the content. Contain the following element: MIMEType	
MIMEType	E6	NM/TM	0..N	Indicates the MIMEType of the script used within the Rich Media content. If not present is indicates that the content does not contain any script.	String
Compression	E5	NO/ TM	1	Indicates whether the delivered rich media scene is compressed/encoded or not. Contains the following attribute: contentEncoding	
content Encoding	A	NM/ TM	1	Scheme used to encode/compress the rich media content 0- None 1- XML 2- Gzip 3- LAsER binary syntax 4- BiM 5-127 reserved for future use 128-255 reserved for proprietary use Note: default value is 0.	unsignedByte
BandwidthRequirement	E1	NO/ TM	0..1	Specification of needed network bandwidth in kbit/s to the access channel described in this fragment. A broadcast service can include multiple accessible streams (same content) with different bandwidth, so that the terminal can make a choice depending on its current reception condition.	unsignedInt
ServiceClass	E1	NM/ TM	1	The ServiceClass identifies the class of service. This identification is more detailed than the 'ServiceType' element in the 'Service' fragment and allows the association of service / access combination to specific applications. The Terminal SHALL be able to interpret the ServiceClasses as defined in OMNA registry (see Appendix E).	string

ReferredSGInfo	E1	NO/TM	0..1	<p>This element is used in the context of Service Guide within Service Guide functionality (see section 5.2 of this document), and specifies additional information related to the referred Service Guide. for which the delivery session is declared by this 'Access' fragment</p> <p>This element SHALL be present when the value of 'ServiceClass' is "urn:oma:bcas:oma_bsc:csg:1.0" or "urn:oma:bcas:oma_bsc:sg:1.0" and SHALL be absent otherwise.</p> <p>Contains the following elements:</p> <p>BSMSelector Service</p>	
BSMSelector	E2	NM/TM	0..N	<p>Specifies the BSM associated with the referred Service Guide.</p> <p>Contains the following attribute:</p> <p>idRef</p>	
idRef	A	NM/TM	1	<p>Reference to the identifier of the BSMSelector declared within the 'BSMList' in the ServiceGuideDeliverDescriptor. For information regarding the scope of this identifier in the context of the Service Guide within Service Guide functionality, refer to section 5.2 of this document.</p>	anyURI
Service	E2	NM/TM	0..N	<p>This elements declares the id of the Service fragment in the referring Service Guide, for which the referred Service Guide declared in this 'Access' fragment provides additional fragments.</p> <p>Note that a referred Service Guide can complement more than one service in the referring Service Guide.</p> <p>This element SHALL only be instantiated in the complementing case, i.e. when the value of 'ServiceClass' is "urn:oma:bcas:oma_bsc:csg:1.0".</p>	anyURI
PreviewDataReference	E1	NM/TM	0..N	<p>Reference to the 'PreviewData' fragment which specifies the preview data (Eg. picture, video clip, or low-bit rate stream).associated with this access.</p> <p>It is possible that there are more than one PreviewDataReference instances associated with the same fragment, in which case, the values of "usage" attributes of these PreviewDataReference instances SHALL be different.</p> <p>Contains the following attributes:</p> <p>idRef usage</p>	
idRef	A	NM/TM	1	<p>Identification of the 'PreviewData' fragment which this fragment associated with.</p>	anyURI



usage	A	NM/ TM	1	Specifies the usage of the associated preview data. Possible values: 0. unspecified 1. Service-by-Service Switching 2. Service Guide Browsing 3. Service Preview 4. Barker 5. Alternative to blackout 6-127. reserved for future use 128-255. reserved for proprietary use The explanation and limitation on the above preview data usages is specified in section 5.7.	unsignedByte
-------	---	-----------	---	--	--------------

Notification Reception	E1	NO/TO	0..1	<p>Reception information for Notification Messages.</p> <p>In case of delivery over Broadcast channel, 'IPBroadcastDelivery' specifies the address information for receiving service-specific Notification messages.</p> <p>In case of delivery over Interaction channel, 'PollURL' specifies address information for polling service-specific Notification messages, and 'PollPeriod' specifies the associated polling period for service-specific Notification messages and independent Notification services.</p> <p>If this element is present, at least one of the elements "IPBroadcastDelivery", "PollURL" or "PollPeriod" SHALL be present.</p> <p>For independent Notification services, i.e. when the "ServiceClass" element of the "Access" fragment is equal to the value "urn:oma:bcast:oma_bsc:nt:1.0" and the referenced service is of type "Notification" (i.e. "ServiceType" element value equals to "7" in the "Service" fragment), the following applies: The "NotificationReception" element MAY be instantiated to signal the polling period via the "PollPeriod" sub-element, but it SHALL NOT contain the "IPBroadcastDelivery" and "PollURL" elements as the transport information is signalled in the "AccessType" mandatory element.</p> <p>If multiple unicast delivery entries are declared in this "Access" fragment, the terminal SHALL balance polling requests, within the whole set of entries. Further, after having selected randomly, at a given time, a given URL entry for polling, the terminal SHOULD use it for a while to benefit from cache management information as specified in HTTP 1.1 [RFC 2616], as it is reminded that this information is scoped to one given URL. This element SHALL be supported by the Network in case it supports the Notification function. Similarly, this element SHALL be supported by the Terminal in case it supports the Notification function.</p> <p>Contains the following elements:                  IPBroadcastDelivery                  PollURL</p>	
------------------------	----	-------	------	--	--

IPBroadcast Delivery	E2	NM/TM	0..1	<p>Provides IP multicast address and port number for reception of Notification Messages over the broadcast channel.</p> <p>The ‘port’ is MANDATORY in both Network and Terminal because a designated UDP Port has to be used to deliver the Notification Message through an on-going session or the designated session while the ‘address’ is OPTIONAL to be used for the delivery of Notification Messages through the designated multicast or broadcast session.</p> <p>In case the ‘address’ attribute is not provided the destination address provided by this access fragment SHALL be assumed.</p> <p>Contains the following attributes: port address</p>	
port	A	NM/TM	1	Service-specific Notification Message delivery UDP destination port number, delivery over broadcast channel.	unsignedInt
address	A	NM/TM	0..1	Service-specific Notification Message delivery IP multicast address, delivery over broadcast channel.	string
PollURL	E2	NM/TM	0..N	<p>URL through which the terminal can poll service-specific Notification Messages.</p> <p>If there are multiple instances of “PollURL” signaled, the terminal SHALL balance polling requests, within the set of “PollURL”. Further, after having selected randomly, at a given time, a given URL, the terminal SHOULD use it for a while to benefit from cache management information as specified in HTTP 1.1 [RFC 2616], as it is reminded that this information is scoped to one given URL.</p>	anyURI
PollPeriod	E2	NO/TM	0..1	<p>This element specifies the polling interval for service-specific Notification Messages or independent Notification services. The NTC is expected to poll for notification messages every “PollPeriod” seconds.</p> <p>When this attribute is instantiated, no caching mechanisms of HTTP 1.1 [RFC 2616] SHOULD conflict with the fact that the NTC is expected to request for a fresh set of Notification Messages every “PollPeriod” value.</p> <p>The unit of this attribute is seconds</p>	decimal

SmartcardProvisioningReception	E1	NO/TO	0..1	<p>Reception information for Smartcard provisioning messages reception on the broadcast channel. This element is used to define all elements necessary for the Reception of the message. In this element, identification of the Subscriber group of the concerned Smartcards is specified. The technology used by the terminal to transfer data to the Smartcard is also defined.</p> <p>This element SHALL be present when the value of 'ServiceClass' is "urn:oma:bcast:oma_bsc;sp:1.1" and SHALL be absent otherwise.</p> <p>Contains the following elements:</p> <p>Addressing SmartcardAccess</p>	
Addressing	E2	NO/TM	0..1	<p>This element defines the addressing scheme used and the SubscriberGroupIdentifier used to target unique Smartcard or subset of Smartcards within a group.</p> <p>When this element is absent, all Smartcards are targeted and the terminal SHALL NOT perform address filtering</p> <p>Contains the following attributes:</p> <p>type groupSize</p> <p>Contains the following element:</p> <p>SubscriberGroupIdentifier</p>	
type	A	NM/TM	1	<p>This element defines the type of addressing scheme used for the Smartcard provisioning.</p> <p>0: Fixed Group Address Mode (as defined in [BCAST12-Services], in section 5.19.3).</p> <p>1-127: Reserved for future use</p>	unsignedByte
groupSize	A	NM/TM	1	<p>Indicates the size of the groups. Value "n" indicates a group of 2<sup>n</sup> Smartcards.</p> <p>The use of this element for the filtering is described in [BCAST12-Services].</p>	unsignedByte
SubscriberGroupIdentifier	E3	NM/TM	1..N	<p>This element defines the Subscriber Group concerned by the Smartcard provisioning service. The Terminal uses this information to process only messages targeting the user's Smartcard and to discard messages the Smartcard is not concerned to.</p> <p>for Addressing Type equals to "0- Fixed Group Address Mode": ComplexType defined in section 5.19.3 [BCAST12-Services]</p> <p>The use of this element for the filtering is described in [BCAST12-Services].</p>	complexType

SmartcardAccess	E2	NM/TM	1	This element defines the technology used to transfer messages to the card. This element is used as terminal capabilities required to consume the service. Contains the following attribute: technology Contains the following elements: URL Request	
technology	A	NM/TM	1	Specifies the technology type used to transfer the file to the Smartcard  0 – Envelope (as defined in [ETSI TS 102221]) 1 – SCWS (as defined in [OMA-TS-Smartcard_Web_Server-V1_1]). 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
URL	E3	NO/TM	0..1	This element specifies the URL used to transfer the file to the SCWS in case of Technology attribute equals to “1- SCWS”. This element SHALL be present in case of Technology attribute equals to “1- SCWS”. This element is absent in case of ‘technology’ attribute equals to “0- Envelope”.	anyURI
Request	E3	NO/TM	0..1	This element specifies the request used to transfer the file to the SCWS in case of Technology attribute equals to “1- SCWS”. This element SHALL be present in case of Technology attribute equals to “1- SCWS”. This element is absent in case of ‘technology’ attribute equals to “0- Envelope”. Contains the following attribute type	
type	A	NM/TM	1	Specifies the Request type used to transfer the file to the SCWS: 0- PUT 1- 127 reserved for future use 128 – 255 reserved for proprietary use	unsignedByte
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.1.2.5 Session Description Information

The Session Description information MAY be directly contained in the associated ‘Access’ fragment. Further, such information, in the form of ‘SessionDescription’ fragments MAY be referenced by ‘idRef’ attribute in the associated ‘Access’ fragment, and encapsulated into a SGDU for broadcast delivery. Finally, ‘SessionDescription’ fragments MAY be referenced from the associated ‘Access’ fragment by the ‘uri’ attribute containing an absolute URI pointing to an external resource which is fetched via the interaction channel.

### 5.1.2.5.1 SessionDescription fragment

An 'Access' fragment MAY reference zero or more 'SessionDescription' fragments by reference inside the 'SessionDescription' element in the following manner:

- via the SDPRef element, the 'Access' fragment MAY reference zero or one 'SessionDescription' fragments, containing Session Description information, formatted according to the syntax of Session Description Protocol (SDP) [RFC 4566], or
- via the USBDef element, the 'Access' fragment MAY reference zero or one 'SessionDescription' fragments containing MBMS User Service Bundle Descriptions (USBD) as specified in [3GPP TS 26.346] section 5.2.2, or
- via the ADPRef element, the 'Access' fragment MAY reference zero or one 'SessionDescription' fragments containing associated delivery procedure description information as specified in [BCAST12-Distribution] section 5.3.4., formatted in XML.

The following rules are applicable to associated delivery procedures:

- In case of associated delivery procedures related to file delivery (section 5.3 of [BCAST12-Distribution]), in the context of one 'SessionDescription' instance
  - If an MBMS User Service Bundle Description (USBD) is provided, the Associated Delivery Procedure description MAY be provided as part of the USBD, but SHALL NOT be provided through the 'ADPRef' element.
  - If an 'SDP' or an 'SDPRef' element is provided, then the Associated Delivery Procedure description MAY be provided through the 'ADPRef' element, but SHALL NOT be provided as part of an MBMS USBD.
- In case of associated delivery procedures related to stream delivery (section 6.3 of [BCAST12-Distribution])
  - The Network MAY provide an Associated Delivery Procedure description either
    - conformant with section 6.3.1 of [BCAST12-Distribution], through the 'ADPRef' element, or
    - conformant with [3GPP TS 26.346] and embedded in an MBMS USBD
    - but SHALL NOT provide both in the same 'UnicastServiceDelivery' instance of an 'Access' fragment

The following rules are applicable to file delivery sessions over the interaction channel:

- In order to declare such session, one or more instances of the 'UnicastServiceDelivery' element SHALL be provided. Each of those instance SHALL be provided according to one of the following options:
  - either along with an 'AccessServerURL' element and the 'type' attribute set to 0, 1, or 2, or
  - along with a 'SessionDescription' element and the 'type' attribute set to 6.
- An 'Access' fragment MAY provide more than one file delivery method over the interaction channel through the use of multiples instances of the 'UnicastServiceDelivery' element.

The following rules are applicable to stream delivery sessions over the interaction channel:

- In case a 'UnicastServiceDelivery' element provides several instances of the 'AccessServerURL' element, those instances SHALL provide URLs with the same protocol scheme.
- In case a 'UnicastServiceDelivery' element provides a 'SessionDescription' element, and one or more instances of the 'AccessServerURL' that use the 'rtsp' protocol scheme, no assumption is made regarding how the terminal selects one of the two signalling methods. The network SHOULD ensure that the signalling provided through the 'AccessServerURL' and the 'SessionDescription' element will lead to the same terminal behaviour.

The following applies to the SDP-formatted 'SessionDescription' fragment.

- For IPv6 support in SDP, RFC 3266 is used.
- For IPv4 support in SDP, RFC 4566 is used.
- 'SessionDescription' fragments MAY also contain other SDP extensions.
- If the SDP contains multiple media sections with the same media identifier (e.g. m=audio... or m=video...), then the selection between the media section to be used by the terminal SHALL be according to section 7.2.1.

The Session Description information SHALL be provided using either syntax of SDP in text format, or through a 3GPP MBMS User Service Bundle Description (USB) [3GPP TS 26.346]. BCAST device SHALL support SDP and MAY support USB. The MBMS USB as used in BCAST SHALL contain zero or one reference to FEC Repair Stream Description and SHALL contain one UserServiceDescription (USD). Each USD SHALL refer to one or more DeliveryMethods (DM), and SHALL contain zero or more accessGroup elements (those containing accessBearer elements). Each DM SHALL refer to zero or one AssociatedDeliveryProcedureDescriptions, and SHALL refer to one SessionDescription, and SHALL NOT refer to any SecurityDescription.

### 5.1.2.5.2 Session Description for broadcast streamed media session

The SessionDescription SHALL provide the following parameters:

- Destination IP address and port number for each media in the session
- The start time and end time of the session
- The transport protocol used
- Media types and media formats
- Data rates using SDP bandwidth modifiers

Additionally, the Session Description MAY provide the following parameters:

- The sender IP address
- The mode of MBMS bearer per media
- FEC configuration and related parameters
- Initial buffering delay, using the 'min-buffer-time' attribute as specified in [ETSI 102 472] section 5.3.4.
- Service protection parameters as defined in [BCAST12-ServContProt]
- Declaration of subtitling / closed captioning parameters

For the above parameters, either SDP [RFC 4566] or 'SessionDescription' fragments in MBMS user service description of MBMS User Service Bundle Description (MBMS-USB) [3GPP TS 26.346], or [ETSI 102 472] section 5.2 SHALL be used to describe a broadcast streamed media session. An example is given below. If subtitling is provided the SDP signaling as defined in [RFC 4396] SHALL be used and the Terminal SHALL be able to interpret the signaling.

Each media line (m=<media> <port> <transport> <fmt list>) in the SDP indicates a payload type which is expressed using a <fmt list> sub-field. If the type definition specifies mandatory parameters, these MUST be included in the payload type associated attributes. Optional parameters containing information that can be used to determine as to whether the decoder installed in the terminal can decode the stream/content SHOULD be included in the payload type associated attributes

SDP Example:

```
v=0
o=- 424 3292855200 IN IP6 FF15:0:0:0:0:0:81:1BC
s=Unencrypted Mobile TV Example
c=IN IP6 FF15:0:0:0:0:0:81:1BD
t=0 0
m=audio 49172 RTP/AVP 96
b=AS:64
a=rtpmap:96 mpeg4-generic/32000
a=fmtp:96 streamtype=5; profile-level-id=15; mode=AAC-hbr;
config=1290; SizeLength=13; IndexLength=3; IndexDeltaLength=3;
Profile=1;
m=video 49170 RTP/AVP 97
b=AS:250
a=rtpmap:97 H264/90000
a=fmtp:97 profile-level-id=42c00d; packetization-mode=1;sprop-
parameter-sets=Z0LADZtAoPiA,aN4liA==;
m=video 49174 RTP/AVP 98
```

```
a=rtpmap:98 3gpp-tt/1000
a=fmtp:98 sver=60;tx=20;ty=200;width=200;height=50;max-w=720;max-h=576
```

### 5.1.2.5.3 Session Description for broadcast file delivery session

The ALC specification [RFC 3450] describes required and optional parameters for an ALC session and media descriptions. The FLUTE specification [RFC 3926] also states a set of required and optional parameters for describing a FLUTE session. This section specifies the Session Description using SDP for both ALC and FLUTE sessions that is used for file distribution sessions. The formal specification of the parameters is given in ABNF [RFC 2234].

#### 5.1.2.5.3.1 Session Descriptors for FLUTE Sessions

The Session Description for FLUTE sessions SHALL provide the following parameters:

- Sender IP address
- The number of channels in the session
- Destination IP address and port number for each channel in the session, given in the corresponding media line
- The Transport Session Identifier of the session
- The file delivery protocol ID
- The start time and end time of the session

Additionally, the Session Description MAY provide the following parameters:

- The mode of MBMS bearer per media
- FEC configuration and related parameters
- Data rates using SDP bandwidth modifiers

The parameters SHALL be formatted according to the definitions in section 7.3 of [3GPP TS 26.346] or section 6.1.13 of [ETSI 102 472].

The Session Description MAY be referenced in an MBMS User Service Bundle Description (MBMS-USBD) [3GPP TS 26.346].

#### Example of FLUTE Session Description:

```
v=0
o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24
s= Example of file delivery session description using FLUTE
i=More information
t=2873397496 2873404696
a=FEC-declaration:0 encoding-id=0;
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
a=flute-tsi:3
a=flute-ch:1
m=application 12345 FLUTE/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E24/1
b=AS:64
a=FEC:0
```

#### 5.1.2.5.3.2 Session Descriptors for ALC Sessions

The session parameters defined for FLUTE sessions SHALL also apply for ALC sessions. For these parameters, SDP [RFC 4566] SHALL be used.

The Session Description MAY be referenced in an MBMS User Service Bundle Description (MBMS-USBD) [3GPP TS 26.346].



The parameters for ALC sessions SHALL be formatted according to the following syntax and semantics.

### Sender IP address

There SHALL be exactly one IP sender address per file distribution session, and thus there SHALL be exactly one IP source address per complete file distribution session SDP description. The source IP address SHALL be provided using a 'source-filter' attribute, which has the following syntax in ABNF:

```
source-filter="a=source-filter: incl IN" SP addr-type SP dest-address SP src-list
    addr-type= "IP4" | "IP6"
    dest-address="*"
    src-list = unicast-address
```

unicast-address is an IP4 or IP6 address depending on addr-type.

The following exceptions apply to the source-filter:

- Exactly one source address MAY be specified by this attribute such that exactly one source address is given by the src-list field.
- There SHALL be exactly one 'source-filter' attribute per complete file distribution session SDP description, and this SHALL be in the session part of the Session Description (i.e., not per media).

Note that the destination address is given as "\*", which indicates that the source filter applies to all destination addresses.

### Number of channels

Multiple channels MAY be used for several purposes. Receivers that are capable of receiving multiple channels simultaneously can benefit from the delivery of urgent files over multiple channels to receive the file at a faster rate. Multiple channels can also be used to deliver files with different FEC encodings, so that receivers can select the FEC code that they support to receive the file.

The multiple channel attribute parameter indicates to the receiver the number of channels the sender is using in the ALC session to transmit data. The value specified by this descriptor MAY be used by the receiver to check consistency of the SDP by counting the number of *m*-lines describing the destinations.

The number of channels is given in SDP syntax for ALC sessions as follows:

```
alc-channel-line = "a=alc-ch:" ch CRLF
ch = integer; integer is defined in [ABNF]
```

where *ch* is the number of channels used by the sender to transmit data in a file distribution session.

### Destination IP Address and Port Number per Channel, and Media and Format List

Each channel SHALL be described by the media-level channel descriptor. These channel parameters SHALL be per channel:

- IP destination address
- Destination port number.

The IP destination address SHALL be defined according to the "connection data" field ("c=") of SDP [RFC 4566]. The destination port number SHALL be defined according to the <port> sub-field of the media announcement field ("m=") of SDP. The media announcement field is specified in [RFC 4566] as follows:

```
m=media SP port ["/" integer] SP proto 1*(SP fmt) CRLF
```

The protocol identifier, which is given by the proto sub-field, SHALL be ALC/UDP for ALC sessions. The media sub-field SHALL be set to “application” and the format list SHALL be set to “0” to indicate that formats are not in use. Only one port SHALL be defined per media-line, so that there SHALL be exactly one media-line per channel.

The presence of an ALC session on a certain channel SHALL be indicated by using the ‘*m*-line’ in the SDP description as shown in the following example:

```
m=application 12345 ALC/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
```

Note that the above destination address is an IPv6 multicast address.

### Transport Session Identifier (TSI)

The combination of the TSI and the IP source address identifies the ALC session. Each TSI SHALL uniquely identify an ALC session for a given IP source address during the time that the session is active and also for a large enough time before and after the active session time.

The TSI SHALL be defined according to the SDP attribute given below. There SHALL be exactly one occurrence of this descriptor in a complete SDP Session Description and it SHALL appear at session level.

The syntax for an ALC session is given below in ABNF format:

```
alc-tsi-line = "a=alc-tsi:" tsi CRLF
tsi=1 *DIGIT
```

Where *tsi* gives the TSI.

### Session Timing Parameters

A file distribution session start and end times SHALL be defined according to the SDP timing field (“t=”) [RFC 4566].

### FEC capabilities and related parameters

A FEC-declaration session-level attribute is defined which results in, e.g.:

```
a=FEC-declaration:0 encoding-id=0
```

Several FEC-declaration lines MAY be declared for the same session. The FEC-declaration is OPTIONAL as the information may be available elsewhere (e.g. in the codepoint header field of LCT header). If this attribute is not used and is not available elsewhere, the terminal SHALL assume that support for FEC encoding id 0 (Compact No-Code FEC code) is sufficient capability to enter the session.

A new media-level FEC-reference attribute ‘FEC’ SHALL be defined to refer to the used FEC declaration. It can result in, e.g.;

```
a=FEC:0
```

This attribute is used as a short hand to inherit one of one or more session-level FEC-declarations to a specific media (i.e. channel).

The syntax for the attributes in ABNF [18] is:

```
fec-declaration-line = "a=FEC-declaration:" fec-ref SP fec-enc-id [ ";" SP fec-inst-id ] CRLF
fec-ref = 1 *3DIGIT (value is the SDP-internal identifier for FEC-declaration).
fec-enc-id = "encoding-id=" enc-id
enc-id = 1 *DIGIT (value is the FEC Encoding ID used).
```

```
fec-inst-id = "instance-id=" inst-id
```

```
inst-id = 1 *DIGIT (value is the FEC Instance ID used).
```

```
fec-line = "a=FEC:" fec-ref CRLF
```

The SDP declares the default FEC encoding scheme (on session level). The FEC encoding scheme may however change from file to file and this is overwritten by declarations in the EXT\_FTI ALC/LCT header, or in File Description of the Service Guide.

### Bandwidth Specification

The maximum bit-rate occupied by each channel of the ALC session SHALL be specified using the "AS" bandwidth modifier [RFC 4566] on media level (i.e. for each single channel). The Application Specific (AS) maximum bandwidth for an ALC channel SHALL be the largest sum of the sizes of all packets transmitted during any one second long period, expressed in kilobits. The size of the packet SHALL be the complete packet including the protocol headers, i.e. IP, UDP, ALC/LCT headers and the data payload.

### Example of ALC Session Description

```
v=0
o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24
s=Example of file delivery session description using ALC
i=More information
t=2873397496 2873404696
a=FEC-declaration:0 encoding-id=0;
a=FEC-declaration:1 encoding-id=1;
a=source-filter: incl IN IP6 *2201:056D°:112E:144A:1E24
a=alc-tsi:3
a=alc-ch :2
m=application 12345 ALC/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
b=AS:64
a=FEC:0
m=application 12346 ALC/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E25
b=AS:64
a=FEC:1
```

### 5.1.2.6 Purchase Item

A purchase item groups one or multiple services/contents, or schedules associated with specific services or contents that an end-user can purchase or subscribe to as a whole.

An instance of the 'PurchaseItem' fragment SHALL only reference one type of 'Service', 'Schedule', 'Content', or 'PurchaseItem' fragments. This constraint is expressed by the using the 'choice' element in the Service Guide XML schema.

Name	Type	Category	Cardinality	Description	Data Type
------	------	----------	-------------	-------------	-----------

PurchaseItem	E			<p>‘PurchaseItem’ fragment</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>id</li> <li>version</li> <li>validFrom</li> <li>validTo</li> <li>globalPurchaseItemID</li> <li>binaryPurchaseItemID</li> <li>weight</li> <li>closed</li> </ul> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>ServiceReference</li> <li>ScheduleReference</li> <li>ContentReference</li> <li>PurchaseItemReference</li> <li>ProtectionKeyID</li> <li>Name</li> <li>Description</li> <li>StartTime</li> <li>EndTime</li> <li>ParentalRating</li> <li>DependencyReference</li> <li>ExclusionReference</li> <li>PrivateExt</li> </ul>	
id	A	NM/TM	1	ID of the ‘PurchaseItem’ fragment. The value of this attribute SHALL be globally unique.	anyURI
version	A	NM/TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no validFrom attribute is given.	unsignedInt
validFrom	A	NM/TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp. The validFrom time of the PurchaseItem SHALL be no earlier than the latest of the validFrom time(s) of the referenced PurchaseItem(s).	unsignedInt
validTo	A	NM/TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp. The ‘validTo’ time of the PurchaseItem SHALL be no later than the earliest of the validTo time(s) of the referenced PurchaseItem(s).	unsignedInt
globalPurchaseItemID	A	NM/TM	1	The globally unique identifier identifying the purchase item described by this fragment. The ‘globalPurchaseItemID’ is used when identifying the purchase item during purchase or other purchase item related actions (see [BCAST12-Services]).	anyURI

binaryPurchaseItemID	A	NO/TO	0..1	A 32-bit identifier generated by BSM, identifying the purchase item described by this fragment. The binaryPurchaseItemID is used to relate this purchase item with the corresponding BCRO. Networks and Broadcast Devices implementing the DRM Profile [BCAST12-ServContProt] SHALL support this attribute.	unsignedInt
weight	A	NM/TM	0..1	Intended order of display of this purchase item relative to other purchase items as seen by the end user. The order of display is by increasing weight value (i.e., purchase item with lowest weight is displayed first). Default: 65535	unsignedShort
closed	A	NM/TM	0..1	If present and value = true, it indicates the Purchase Item is closed to new subscribers. If value = false, unspecified. Default: false.	boolean
ServiceReference	E1	NM/TM	0..N	References to the 'Service' fragments which belong to this PurchaseItem. Note: a 'Service' fragment can be referenced by multiple PurchaseItems. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Service' fragment which this 'PurchaseItem' fragment is associated with.	anyURI
ScheduleReference	E1	NM/TM	0..N	References to the 'Schedule' fragments which belong to this PurchaseItem. Note: a 'Schedule' fragment can be referenced by multiple 'PurchaseItem' fragments. Contains the following attribute: idRef Contains the following element: PresentationWindowIDRef	
idRef	A	NM/TM	1	Identification of the 'Schedule' fragment which the 'PurchaseItem' fragment relates to.	anyURI
PresentationWindowIDRef	E2	NM/TM	0..N	Relation reference to the PresentationWindow to which the 'PurchaseItem' fragment belongs. The 'PresentationWindowIDRef' declared in this element SHALL be the complete collection or a subset of the PresentationWindow ids declared in the 'Schedule' fragment, to which the above Schedule 'idRef' reference belongs.	unsignedInt
ContentReference	E1	NM/TM	0..N	References to the 'Content' fragments which belong to this PurchaseItem. Contains the following attribute: idRef  Note: a 'Content' fragment can be referenced by multiple 'PurchaseItem' fragments.	
idRef	A	NM/TM	1	Identification of the 'Content' fragment which this 'PurchaseItem' fragment is associated with.	anyURI

PurchaseItemReference	E1	NM/TM	0..N	<p>References to the 'PurchaseItem' fragments that are included in the purchase or subscription of this PurchaseItem, and thus need not be purchased separately.</p> <p>Note: a 'PurchaseItem' fragment can be referenced by multiple PurchaseItems.</p> <p>The depth of the 'PurchaseItem' tree SHALL NOT be more than three.</p> <p>'PurchaseItem' fragment referencing other 'PurchaseItem' fragments SHALL NOT cause contradictory or circular reference chains.</p> <p>The reference to a PurchaseItem SHALL NOT depend on, nor be excluded by, the subscription to other purchasable PurchaseItem(s), if the latter PurchaseItem(s) are indeed not subscribed.</p> <p>Contains the following attribute: idRef</p>	
idRef	A	NM/TM	1	<p>Identification of the 'PurchaseItem' fragment which this 'PurchaseItem' fragment is associated with.</p>	anyURI
ProtectionKeyID	E1	NO/TO	0..N	<p>Key identifier needed to access protected service/content. This information allows the terminal to determine whether or not it has the correct key material to access service(s)/content item(s) within a PurchaseItem.</p> <p>In a scenario where this fragment is shared among multiple service providers, different key identifiers from the different service providers to access this specific protected service/content item may be mixed in this element and the terminal SHOULD be able to sort out the key identifiers associated with the terminal's affiliated service provider based on the Key Domain ID. How this is used is out of scope and is left to implementation.</p> <p>The network and terminal SHALL support this element in case the Smartcard Profile is supported [BCAST12-ServContProt].</p> <p>The protection key identifiers to access a specific service or content item SHALL only be signalled in one of the following fragment types: 'Service', 'Content', 'PurchaseItem', 'PurchaseData' or 'Access' fragments, but not mixed.</p> <p>Contains the following attribute: type</p>	base64Binary

type	A	NM/TM	1	<p>Type of ProtectionKeyID, possible values:</p> <p>0: ProtectionKeyID is the 5-byte long concatenation of the Key Domain ID with the Key group part of the SEK/PEK ID, where both values are as used in the Smartcard Profile [BCAST12-ServContProt].</p> <p>The Key number part SHALL NOT be provided.</p> <p>The Terminal MAY use the Key Domain ID and Key group part of the ProtectionKeyID to determine whether it already has SEK/PEK applicable to the related service/content item. The Terminal MAY use this information to indicate to the user which services/content items can currently be accessed. The Terminal SHALL not use the SEK/PEK ID in the ProtectionKeyID to request a missing SEK or PEK. It is possible for the Terminal to request missing SEK or PEK based on the information from the secure function after the STKM decryption has been failed.</p> <p>1-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte
				<p><b>Start of program guide</b></p> <p>The program guide elements of this fragment are grouped between the Start of program guide and end of program guide cells in this fragment.</p> <p>The program guide elements are for user interpretation. This enables the content creator to provide user readable information about the service. The terminal SHOULD use all declared program guide elements in this fragment for presentation to the end-user. The terminal MAY offer search, sort etc functionalities.</p> <p>The Program Guide consists of the following elements:</p> <p>Name Description StartTime EndTime ParentalRating Extension</p>	
Name	E1	NM/TM	1..N	Name of the PurchaseItem, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
Description	E1	NM/TM	0..N	Description of the purchase item, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this Element.	string

StartTime	E1	NM/TM	0..1	<p>The first moment when the item can be purchased, which is for presentation purposes to the end user in UTC, using 'dateTime' XML built-in Datatype.</p> <p>yyyy=year mm=month dd=day T = time separator to clarify the different use of 'mm' hh=hour mm = minutes</p>	dateTime
EndTime	E1	NM/TM	0..1	<p>The last moment when the item can be purchased, which is for presentation purposes to the end user in UTC, using 'dateTime' XML built-in datatype.</p> <p>yyyy=year mm=month dd=day T = time separator to clarify the different use of 'mm' hh=hour mm = minutes</p>	dateTime
ParentalRating	E1	NM/TM	0..N	<p>The ParentalRating element defines criteria parents can use to determine whether the associated item is suitable for access by children, defined according to the regulatory requirements of the service area.</p> <p>This determines the rating level for service purchase, not the rating level of the actual service consumption.</p> <p>The terminal SHALL support 'ParentalRating' being a free string, and the terminal MAY support the structured way to express the parental rating level by using the 'ratingSystem' and 'ratingValueName' attributes as defined below.</p> <p>Contains the following attributes: ratingSystem ratingValueName</p>	string



ratingSystem	A	NO/TM	0..1	<p>Specifies the parental rating system in use, in which context the value of the 'ParentalRating' element is semantically defined.</p> <p>This allows terminals to identify the rating system in use in a non-ambiguous manner and act appropriately.</p> <p>This attribute SHALL be instantiated when a rating system is used.</p> <p>Absence of this attribute means that no rating system is used (i.e. the value of the 'ParentalRating' element is to be interpreted as a free string).</p> <p>If this attribute is instantiated:</p> <ul style="list-style-type: none"> <li>– The value of this attribute SHALL be one of the 'rating_type' values as listed in the OMA BCAST Parental Rating System Registry at [OMNA].</li> <li>– The 'ParentalRating' element SHALL contain the string representation of a number that is a valid 'rating_value' in this particular rating system.</li> <li>– This attribute MAY contain the value '10' (OMA BCAST generic rating scheme), allowing to define a rating value in a non-registered parental rating system. In such case, the 'ParentalRating' element SHALL contain the string representation of a number between 1 and 255, 1 being the least and 255 being the most restrictive rating value. As these values are generic, the human-readable label of that rating value SHALL be signalled in the attribute 'ratingValueName'.</li> </ul>	unsignedByte
ratingValueName	A	NO/TM	0..1	<p>The human-readable name of the rating value given by this ParentalRating element.</p> <p>This attribute SHALL be present in case the 'ratingSystem' attribute contains the value '10'.</p>	string
Extension	E1	NM/TM	0..N	<p>Additional information related to this fragment.</p> <p>Contains the following attribute: url</p> <p>Contains the following element: Description</p>	
url	A	NM/TM	1	URL containing additional information related to this fragment.	anyURI
Description	E2	NM/TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
<b>End of program guide</b>					

Dependency Reference	E1	NO/TO	0..N	Reference to those 'PurchaseItem' fragments, subscription to at least one of which is REQUIRED before subscription SHOULD be offered for this PurchaseItem. The depth of the PurchaseItem tree SHALL NOT be more than three. 'PurchaseItem' fragment referencing other 'PurchaseItem' fragments SHALL NOT cause contradictory or circular reference chains.  Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'PurchaseItem' fragment that is referenced by this element.	anyURI
ExclusionReference	E1	NO/TO	0..N	Reference to those 'PurchaseItem' fragments, subscription to which SHOULD NOT be offered after the user subscribes to this PurchaseItem. The depth of the PurchaseItem tree SHALL NOT be more than three. 'PurchaseItem' fragment referencing other 'PurchaseItem' fragments SHALL NOT cause contradictory or circular reference chains.  Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'PurchaseItem' fragment that is referenced by this element.	anyURI
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

**5.1.2.7 Purchase Data**

The Purchase Data fragment serves the following purposes:

- To express from which purchase channel a certain purchase item can be purchased
- To express all information about a purchase item that is specific for the purchase channel (which can include the price information)

Name	Type	Category	Cardinality	Description	Data Type
------	------	----------	-------------	-------------	-----------

PurchaseData	E			<p>‘PurchaseData’ fragment</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>id</li> <li>version</li> <li>validFrom</li> <li>validTo</li> <li>globalPurchaseDataID</li> </ul> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>ProtectionKeyID</li> <li>Description</li> <li>PriceInfo</li> <li>PromotionInfo</li> <li>Extension</li> <li>Offer Details</li> <li>PurchaseItemReference</li> <li>PurchaseChannelReference</li> <li>PreviewDataReference</li> <li>TermsOfUse</li> <li>DependencyReference</li> <li>PrivateExt</li> </ul>	
id	A	NM/TM	1	ID of the ‘PurchaseData’ fragment. The value of this attribute SHALL be globally unique.	anyURI
version	A	NM/TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no ‘validFrom’ attribute is given.	unsignedInt
validFrom	A	NM/TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NM/TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
globalPurchaseDataID	A	NO/TO	1	The globally unique identifier of the purchase data described by this fragment. The ‘globalPurchaseDataID’ is referenced by a Coupon document during purchase or other purchase item related actions (see [BCAST12-Services]).	anyURI

ProtectionKeyID	E1	NO/TO	0..N	<p>List of key identifiers needed to access protected services/content. This information allows the terminal to determine whether or not it has the correct key material to access service(s)/content item(s) within a PurchaseItem.</p> <p>In a scenario where this fragment is shared among multiple service providers, different key identifiers from the different service providers to access this specific protected service/content may be mixed in this element and the terminal SHOULD be able to sort out the key identifiers associated with the terminal's affiliated service provider based on the Key Domain ID. How this is used is out of scope and is left to implementation.</p> <p>If min and max values are present, these can be used by the terminal to determine whether or not access to the service / content item is possible given the current permissions.</p> <p>The network and terminal SHALL support this element in case the Smartcard Profile is supported [BCAST12-ServContProt].</p> <p>The protection key identifiers to access a specific service or content item SHALL only be signalled in one of the following fragment types: 'Service', 'Content', 'PurchaseItem', 'PurchaseData' or 'Access' fragments, but not mixed.</p> <p>Contains the following attribute:</p> <p>type min max</p>	base64Binary
type	A	NM/TM	1	<p>Type of ProtectionKeyID:</p> <p>0: ProtectionKeyID is the 5-byte long concatenation of the Key Domain ID with the Key group part of the SEK/PEK ID, where both values are as used in the Smartcard Profile [BCAST12-ServContProt].</p> <p>The Key number part SHALL NOT be provided.</p> <p>The Terminal MAY use the Key Domain ID and Key group part of the ProtectionKeyID to determine whether it already has SEK/PEK applicable to the related service/content item. The Terminal MAY use this information to indicate to the user which services/content items can currently be accessed. The Terminal SHALL not use the SEK/PEK ID in the ProtectionKeyID to request a missing SEK or PEK. It is possible for the Terminal to request missing SEK or PEK based on the information from the secure function after the STKM decryption has been failed.</p> <p>1-127 Reserved for future use 128-255 Reserved for proprietary use</p>	unsignedByte

min	A	NM/TM	0..1	Indicates the lower validity value of the key needed to access the service / content. For type 0, corresponds to the value of the lowest timestamp (32 bits) of an STKM needed to access the service / content, as used in the Smartcard Profile [BCAST12-ServContProt]	nonNegative Integer
max	A	NM/TM	0..1	Indicates the higher validity value of the key needed to access the service / content. For type 0, corresponds to the value of the highest timestamp (32 bits) of an STKM needed to access the service / content, as used in the Smartcard Profile [BCAST12-ServContProt].	nonNegative Integer
				<p style="text-align: center;"><b>Start of program guide</b></p> <p>The program guide elements in this fragment are grouped between the Start of program guide and end of program guide cells in this fragment.</p> <p>The program guide elements are for user interpretation. This enables the content creator to provide user readable information about the service. The terminal SHOULD use all declared program guide elements in this fragment for presentation to the end-user. The terminal MAY offer search, sort etc functionalities.</p> <p>The Program Guide consists of the following elements:</p> <ul style="list-style-type: none"> <li>Description</li> <li>PriceInfo</li> <li>PromotionInfo</li> <li>Extension</li> </ul>	

Description	E1	NM/ TM	0..N	<p>Description of the purchase data, possibly in multiple languages. The language is expressed using built-in XML attribute ‘xml:lang’ with this element.</p> <p>The information is expected to indicate the valid start and end times of a given purchase offer (during which the user could make the corresponding purchase). This time interval is expected to be bounded within the period spanned by the attributes ‘validFrom’ and ‘validTo’.</p> <p>Should the referenced purchase item pertain to metered consumption (PPT, PPP or PPV), ‘Description’ is expected to provide the appropriate, corresponding charging related information, for example:</p> <ul style="list-style-type: none"> <li>• Whether the offered method of purchasing is subscription-based, or pertain to per-unit based (by time or by play) consumption;</li> <li>• For subscription-based purchasing, whether the remaining credits at the end of the subscription period will be carried forward to the next subscription period;</li> <li>• For the credit package (specifically comprising user tokens) purchasable for metered consumption of this purchase item, ‘Description’ is expected to inform the user that these credits are transferable for accessing any other purchase item charged by the same type of credits, along with the “exchange rate” information. Depending on the metered consumption semantics of this purchase item, “exchange rate” is intended to inform the user of: <ul style="list-style-type: none"> <li>○ the amount of live (playback) time units exchangeable for the amount of playback (live) time units granted by the credit package associated with this purchase item, or</li> <li>○ the amount of live or playback time units exchangeable for the number of live or playback instances granted by the credit package associated with this purchase item, or</li> <li>○ the number of live or playback instances exchangeable for the amount of live or playback time units granted by the credit package associated with this purchase item.</li> </ul> </li> </ul>	string
-------------	----	-----------	------	---	--------

PriceInfo	E1	NM/ TM	0..1	<p>Specifies the price information of the purchase item associated with this 'PurchaseData' fragment</p> <p>If the price is not given, it will be negotiated with the user as part of the purchase transaction. In this case, the 'PurchaseData' fragment merely reflects that a certain purchase item can be purchased from the purchase channel.</p> <p>Contains the following attribute: subscriptionType</p> <p>Contains the following elements: MonetaryPrice SubscriptionPeriod</p>	
subscription Type	A	NM/ TM	1	<p>If the offered method of purchase of the referenced purchase item is subscription-based, the type of subscription method. Possible values:</p> <p>0 – one-time subscription: the nominal subscription period is given by the 'SubscriptionPeriod' element, after which the subscription is terminated in the absence of explicit re-subscription. The user will be charged for the amount indicated by the 'MonetaryPrice' element.</p> <p>1 – open-ended subscription: the subscription will be valid until the user unsubscribes. In this case the 'SubscriptionPeriod' element, if present, indicates the frequency at which the user will be charged for the amount specified by the 'Price' element.</p> <p>2 – free trial subscription: the subscription is intended to provide the user a one-time only, cost-free access to the referenced purchase item.</p> <p>3 –Token or Count-based: This informs the user that access to the referenced purchase item is acquired in the purchase of a type of credit package as signalled under OfferDetails.</p> <p>Note that in the case of user tokens, inherent transferability of these credits to viewing programs belonging to other purchase items might lead to depletion of the credits for consuming content belonging to this purchase item.</p> <p>4 – 127 Reserved for future use 128-255 Reserved for proprietary use</p> <p>For a definition of (Token) Pay Per Time, (Token) Pay Per View, number of TEKs and Purse see section 6.6.2 and 6.6.5 of [BCAST12-ServContProt].</p>	unsignedByte

MonetaryPrice	E2	NM/TM	0..N	Specifies the monetary value of the price for subscribing to the associated purchase item if 'SubscriptionPeriod' is present, or the monetary value of a credit package if 'OfferDetails' is present (located outside the program guide). Only one 'MonetaryPrice' per currency SHALL be defined. Contains the following attribute: currency	decimal
currency	A	NM/TM	1	Specifies the monetary currency codes defined in ISO 4217 international currency codes.	string
SubscriptionPeriod	E2	NM/TM	0..1	If the offered purchase method is subscription-based, this specifies the time duration of the subscription period of the purchase item referenced by this 'PurchaseData' fragment. e.g. 1 year as P1Y, 2 month as P2M, 1 hour as PT1H, etc.	duration
startTime	A	NM/TM	0..1	Attribute 'startTime' gives the point of time of the beginning of the 'SubscriptionPeriod'. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
PromotionInfo	E1	NO/TO	0..N	Information of the promotion activities/coupons related to the PurchaseItem. If 'TargetUserProfile' sub-element is present, then PromotionInfo can be displayed if the profile of the terminal user meets the requirements described by 'TargetUserProfile'. Contains the following attributes: id validFrom validTo Contains the following elements: Title TargetUserProfile Description PromotionExtension	
id	A	NM/TM	1	Identifier of one certain PromotionInfo, unique for BSM. This id MAY be used in the purchase process to identify the specific promotion	unsignedInt
validFrom	A	NO/TM	0..1	Start of validity; if not given, the start of validity is assumed in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NO/TM	0..1	End of validity; if not given, the end of validity is assumed in the distant future, and the end time can be specified later by updating the object. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
Title	E2	NM/TM	1..N	Title of the PromotionInfo. The instances of Title element differ only in language. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string



TargetUserProfile	E2	NO/TO	0..N	<p>Profile attributes of the users whom the information of the promotion activity/coupon is targeting at. The detailed personal attribute names and the corresponding values are specified by attributes of ‘attributeName’ and ‘attributeValue’. Amongst the possible profile attribute names are age, gender, occupation, etc. (subject to national/local rules &amp; regulations, if present and as applicable regarding use of personal profiling information and personal data privacy.).</p> <p>The extensible list of ‘attributeName’ and ‘attributeValue’ pairs for a particular PromotionInfo enables end user profile filtering and end user preference filtering of PromotionInfo for the PurchaseData. The terminal SHOULD be able to support ‘TargetUserProfile’ element. The terminal behavior for interpreting and acting upon ‘TargetUserProfile’ is out of the scope of this document.</p> <p>It is RECOMMENDED that use of ‘TargetUserProfile’ element is an “opt-in” capability for users. Terminal settings SHOULD allow users to configure whether to input their personal profile or preference and whether to allow PromotionInfo to be automatically filtered based on the users’ personal attributes without users’ request.</p> <p>Contains the following attributes: attributeName attributeValue</p>	
attributeName	A	NM/TM	1	Profile attribute name.	string
attributeValue	A	NM/TM	1	Profile attribute value.	string
Description	E2	NO/TM	0..N	<p>Description or explanation about the PromotionInfo. The language is expressed using the built-in XML attribute ‘xml:lang’ with this element.</p> <p>Either ‘Description’ or ‘PromotionExtension’ element or both of them SHOULD be specified by the BSM to represent the detailed information on this PromotionInfo.</p>	string
PromotionExtension	E2	NO/TM	0..N	<p>Additional detailed promotional information (e.g. information about coupon sponsors, server location for purchases by using coupons).</p> <p>Either ‘Description’ or ‘PromotionExtension’ element or both of them SHOULD be specified by the BSM to represent the detailed information on this PromotionInfo.</p> <p>Contains the following attribute: url</p> <p>Contains the following element: Description</p>	

url	A	NM/ TM	1	URL containing additional information related to this promotion.	anyURI
Description	E3	NO/ TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
Extension	E1	NO/ TM	0..N	Additional information related to this fragment. Contains the following attribute: url Contains the following element: Description	
url	A	NM/ TM	1	URL containing additional information related to this fragment.	anyURI
Description	E2	NO /TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
<b>End of program guide</b>					
OfferDetails	E1	NO/TO	0..1	This information is used by the terminal to either construct the "Token Purchase Request" message, for the purchase of a token-based credit package, or it indicates the use of the "Service Request" message for the purchase of a count-based credit package. Both messages are defined in [BCAST12-Services]. The description of the 'CreditPackageType' element below provides further information on the different types of credit packages. Instead of being provided for direct display to the user, this info is expected to be translated by the terminal to present to the user meaningful credit package based purchasing information. Contains the following elements: CreditPackageType TotalNumberTokenCredits TotalNumberCountCredits	

CreditPackageType	E2	NM/TM	1	<p>Specifies the type of credit package, associated with PPP, PPT or PPV token or count-based consumption, available for purchase of the purchase item referenced by this 'PurchaseData' fragment.</p> <p>Allowed values are:</p> <ul style="list-style-type: none"> <li>0 – unspecified</li> <li>1 – ServiceTokenPPTLive</li> <li>2 – ServiceTokenPPTPlayback</li> <li>3 - UserTokenPPTLive</li> <li>4 - UserTokenPPT Playback</li> <li>5 - UserTokenPPVLive</li> <li>6 – UserTokenPPPPlayback</li> <li>7 – fixed number of recorded content playbacks</li> <li>8 – fixed time duration for live content consumption – credit carryover disallowed</li> <li>9 – fixed time duration for live content consumption – credit carryover allowed</li> <li>10 – fixed time duration for recorded content consumption</li> <li>11 – unlimited duration for recorded content consumption</li> <li>12 - 127 reserved for future use</li> <li>128 -255 reserved for proprietary use</li> </ul> <p>Contains the attribute: extraTokensPurchaseable</p> <p>For 'CreditPackageType' = 1 or 2 or 3 or 4 or 5 or 6, the element 'TotalNumberTokenCredits' SHALL be instantiated and the element 'TotalNumberCountCredits' SHALL NOT be instantiated.</p> <p>For 'CreditPackageType' = 7 or 8 or 9 or 10, the element 'TotalNumberTokenCredits' SHALL NOT be instantiated and the element 'TotalNumberCountCredits' SHALL be instantiated.</p> <p>For 'CreditPackageType' = 11, the element 'TotalNumberTokenCredits' SHALL NOT be instantiated and the element 'TotalNumberCountCredits' SHALL NOT be instantiated.</p> <p>Note that 'TotalNumberTokenCredits' and 'TotalNumberCountCredits' are mutually exclusive and this is realized in the XML schema by using the &lt;choice&gt; element.</p> <p>Notes (the following information is not intended for user viewing):</p> <ul style="list-style-type: none"> <li>– 'ServiceTokenPPTLive' implies time-based consumption of live content metered by service tokens. For the Smartcard profile, it corresponds to SPE=0x00 in [BCAST12-ServContProt].</li> <li>– 'ServiceTokenPPTPlayback' implies time-based consumption of recorded content metered by service tokens. For the Smartcard profile, it corresponds to SPE=0x01 in [BCAST12-ServContProt].</li> <li>– 'UserTokenPPTLive' implies time-based</li> </ul>	unsignedByte
-------------------	----	-------	---	--	--------------

extraTokensPurchaseable	A	NO/TM	0..1	<p>Indicates for the offered token-based credit package whether or not extra tokens, of quantity 'TotalNumberTokenCredits', can be purchased. Value = "true" means extra tokens can be purchased, and value = "false" means no extra tokens can be purchased. Furthermore, the following rules apply:</p> <ul style="list-style-type: none"> <li>For 'CreditPackageType' = 1, the extra tokens are associated with the live_ppt_purse that is linked to the SEK/PEK ID Key Group;</li> <li>For 'CreditPackageType' = 2, the extra tokens are associated with the playback_ppt_purse that is linked to the SEK/PEK ID Key Group;</li> <li>For 'CreditPackageType' = 3 or 4 or 5 or 6, the extra tokens are associated with the user_purse.</li> </ul>	boolean
TotalNumberTokenCredits	E2	NM/TM	0..1	<p>Specifies the number of token-based credits, associated with PPT, PPP or PPV content, which can be acquired as a token-based credit package unit for consumption of the associated purchase item.</p> <p>Contains the following attributes:  creditType  consumptionAmount  consumptionUnit  maxReplay</p> <p>When 'consumptionUnit' = 3, the value of this element divided by the value of 'consumptionAmount' represents the number of token-based credits consumed per instance of play.</p> <p>When 'consumptionUnit' = 0, 1 or 2, the value of this element divided by the value of 'consumptionAmount' represents the number of token-based credits consumed per time unit of play.</p> <p>Note: the value of 'MonetaryPrice' divided by the value of 'TotalNumberTokenCredits' is the cost per token-based credit. The terminal can calculate and display this information to the user if necessary.</p> <p>Note: the value of 'MonetaryPrice' divided by the value of 'consumptionAmount' is the cost per time unit of play or the cost per instance of play. The terminal can calculate and display this information to the user if necessary.</p>	unsignedShort

creditType	A	NM/ TM	1	<p>Specifies the type of token-based credit. Possible values are:</p> <ul style="list-style-type: none"> <li>0 – unspecified</li> <li>1 – tokens for the DRM Profile</li> <li>2 – service tokens for the Smartcard Profile added to live_ppt_purse (“CreditPackageType” is set to 1 (‘ServiceTokenPPTLive’))</li> <li>3 – service tokens for the Smartcard Profile added to playback_ppt_purse (“CreditPackageType” is set to 2 (‘ServiceTokenPPTPlayback’))</li> <li>4 – user tokens for the Smartcard Profile (“CreditPackageType” is set to 3 (‘UserTokenPPTLive’) or 4 (‘UserTokenPPTPlayback’) or 5 (‘UserTokenPPVLive’) or 6 (‘UserTokenPPPPlayback’))</li> <li>5 – 127 – reserved for future use</li> <li>128-255 – reserved for proprietary use</li> </ul> <p>For a definition of and related processing of (Token) Pay Per Time, (Token) Pay Per View, number of TEKs and Purse see sections 6.6.4, 6.6.7 and 6.6.8 of [BCAST12-ServContProt].</p>	unsignedByte
consumption Amount	A	NM/TM	0..1	<p>Represents the amount of permitted consumption corresponding to this credits package.</p> <p>When ‘consumptionUnit’ = 0 or 1 or 2, this provides the total viewing time corresponding to the total number of token-based credits specified by ‘TotalNumberTokenCredits’.</p> <p>When ‘consumptionUnit’ = 3, this provides the total number of plays or views corresponding to the total number of token-based credits specified by ‘TotalNumberTokenCredits’.</p> <p>This attribute SHALL be present when the amount of permitted consumption corresponding to this token-based credit package is a fixed number. When that is not the case, due to variable cost per play instance or per time unit, as expressed in the LTKM for a given SEK / PEK ID, this attribute SHALL NOT be present.</p>	unsignedShort

consumption Unit	A	NM/TM	1	<p>Describes the type of consumption unit.</p> <p>Allowed values are:</p> <p>0 – time in seconds, used when ‘CreditPackageType’ is set to 1 (‘ServiceTokenPPTLive’) or 2 (‘ServiceTokenPPTPlayback’) or 3 (‘UserTokenPPTLive’) or 4 (‘UserTokenPPTPlayback’)</p> <p>1 – time in minutes, used when ‘CreditPackageType’ is set to 1 (‘ServiceTokenPPTLive’) or 2 (‘ServiceTokenPPTPlayback’) or 3 (‘UserTokenPPTLive’) or 4 (‘UserTokenPPTPlayback’)</p> <p>2 – time in hours, used when ‘CreditPackageType’ is set to 1 (‘ServiceTokenPPTLive’) or 2 (‘ServiceTokenPPTPlayback’) or 3 (‘UserTokenPPTLive’) or 4 (‘UserTokenPPTPlayback’)</p> <p>3 – number of plays, used when ‘CreditPackageType’ is set to 5 (‘UserTokenPPVLive’) or 6 (‘UserTokenPPPPlayback’)</p> <p>4 -127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
maxReplay	A	NM/TM	0..1	<p>This attribute indicates the maximum number or time of plays for this purchase item, as an integer multiple of ‘consumptionAmount’. It informs the terminal of the practical limit on the number of token-based credit packages that can be requested in the Token Purchase Request message (as indicated by the value of ‘purchaseUnitNum’) defined in section 5.1.5.5.1 of [BCAST12-Services]. How this is managed by the BSM is out of scope.</p> <p>If this attribute is not present, it means there is no limitation on the number or time of plays that can be purchased.</p>	unsignedShort

TotalNumberCountCredits	E2	NM/TM	0..1	<p>Specifies the number of count-based credits, associated with PPT or PPP content, which can be acquired as a count-based credit package unit for consumption of the associated purchase item.</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>consumptionAmount</li> <li>consumptionUnit</li> </ul> <p>When 'consumptionUnit' = 3, the value of this element divided by the value of 'consumptionAmount' represents the number of count-based credits consumed per instance of play.</p> <p>When 'consumptionUnit' = 0, 1 or 2, the value of this element divided by the value of 'consumptionAmount' represents the number of count-based credits consumed per time unit of play.</p> <p>Note: the value of 'MonetaryPrice' divided by the value of 'TotalNumberCountCredits' is the cost per count-based credit. The terminal can calculate and display this information to the user if necessary.</p> <p>Note: the value of 'MonetaryPrice' divided by the value of 'consumptionAmount' is the cost per time unit of play or the cost per instance of play. The terminal can calculate and display this information to the user if necessary.</p>	unsignedShort
consumptionAmount	A	NM/TM	1	<p>Represents the amount of permitted consumption corresponding to this credits package.</p> <p>When 'consumptionUnit' = 0 or 1 or 2, this provides the total viewing time corresponding to the total number of count-based credits specified by 'TotalNumberCountCredits'.</p> <p>When 'consumptionUnit' = 3, this provides the total number of plays or views corresponding to the total number of count-based credits specified by 'TotalNumberCountCredits'.</p>	unsignedShort

consumption Unit	A	NM/TM	1	<p>Describes the type of consumption unit.</p> <p>Allowed values are:</p> <p>0 – time in seconds, used when ‘CreditPackageType’ is set to 8 (‘fixed time duration for live content consumption – credit carryover disallowed’) or 9 (‘fixed time duration for live content consumption – credit carryover allowed’) or 10 (‘fixed time duration for recorded content consumption’)</p> <p>1 – time in minutes, used when ‘CreditPackageType’ is set to 8 (‘fixed time duration for live content consumption – credit carryover disallowed’) or 9 (‘fixed time duration for live content consumption – credit carryover allowed’) or 10 (‘fixed time duration for recorded content consumption’)</p> <p>2 – time in hours, used when ‘CreditPackageType’ is set to 8 (‘fixed time duration for live content consumption – credit carryover disallowed’) or 9 (‘fixed time duration for live content consumption – credit carryover allowed’) or 10 (‘fixed time duration for recorded content consumption’)</p> <p>3 – number of plays, used when ‘CreditPackageType’ is set to 7 (‘fixed number of recorded content playbacks’)</p> <p>4 -127 reserved for future use 128-255 reserved for proprietary use</p>	unsignedByte
PurchaseItemReference	E1	NM/TM	1	<p>The PurchaseItem to which this PurchaseData applies to.</p> <p>Contains the following attribute:</p> <p>idRef</p>	
idRef	A	NM/TM	1	<p>Identification of the ‘PurchaseItem’ fragment which this ‘PurchaseData’ fragment is associated with.</p>	anyURI
PurchaseChannelReference	E1	NM/TM	1..N	<p>The PurchaseChannel through which the identified PurchaseItem can be obtained.</p> <p>Contains the following attribute:</p> <p>idRef</p>	
idRef	A	NM/TM	1	<p>Identification of the PurchaseChannel fragment which this ‘PurchaseData’ fragment is associated with.</p>	anyURI



PreviewDataReference	E1	NM/ TM	0..N	<p>Reference to the 'PreviewData' fragment which specifies the preview Data (eg. picture, video clip, or low-bit rate stream).associated with this purchase data.</p> <p>It is possible that there are more than one PreviewDataReference instances associated with the same fragment, in which case, the values of "usage" attributes of these PreviewDataReference instances SHALL be different.</p> <p>Contains the following attributes:</p> <p>idRef usage</p>	
idRef	A	NM/ TM	1	<p>Identification of the 'PreviewData' fragment which this 'PurchaseData' fragment is associated with.</p>	anyURI
usage	A	NM/ TM	1	<p>Specifies the usage of the associated preview data. Possible values:</p> <p>0. unspecified 1. Service-by-Service Switching 2. Service Guide Browsing 3. Service Preview 4. Barker 5. Alternative to blackout 6-127. reserved for future use 128-255. reserved for proprietary use</p> <p>The explanation and limitation on the above preview data usages is specified in section 5.7.</p>	unsignedByte
TermsOfUse	E1	NO/ TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>Contains the following attributes:</p> <p>type id userConsentRequired</p> <p>Contains the following elements:</p> <p>Country Language PreviewDataIDRef TermsOfUseText</p>	

type	A	NM/ TM	1	The way the terminal SHALL interpret the Terms of Use: 0 – Display before purchasing or subscribing. If ‘TermsOfUse’ element of type ‘0’ is present, terminal SHALL render the Terms of Use prior to initiating purchase or subscription request related PurchaseItem associated with this fragment. 1 – Not used. 2 - 127 reserved for future use 128 -255 reserved for proprietary use	unsignedByte
id	A	NM/ TM	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsent Required	A	NM/ TM	1	Signals whether user consent for these Terms of Use is needed. true: User consent is required for these Terms of Use and needs to be confirmed in the subscription / purchase request message related to the PurchaseItem associated with this fragment. false: User consent is not required for the Terms of Use.	boolean
Country	E2	NM/ TM	0..N	List of countries for which the Terms of Use is applicable if purchasing associated PurchaseItem in that country. Each value is a Mobile Country Code according [ITU-MCC]. If this element is omitted, the Terms of Use are applicable to any country.	string of 3 digits
Language	E2	NM/ TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/ TM	0..1	Reference to the ‘PreviewData’ fragment which carries the representation of Terms of Use. If this element is not present, the ‘TermsOfUseText’ element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/ TM	0..1	Terms of Use text to be rendered. If this element is not present, the ‘PreviewDataIDRef’ element SHALL be present (Implementation in XML schema using <choice>).	string

Dependency Reference	E1	NO/TM	0..1	<p>This field identifies a PurchaseData fragment that SHALL be purchased before this PurchaseData is purchased.</p> <p>When the DependencyReference is purchased, the terminal SHALL initiate purchase of this 'PurchaseData'. If this field points recursively to this 'PurchaseData', then this PurchaseData SHALL BE purchased by the BCAST terminal.</p> <p>In either case, if this PurchaseData has nonzero cost then the user MUST be prompted to confirm the purchase.</p> <p>The depth of the PurchaseData tree SHALL NOT be more than three.</p> <p>Contains the following attribute: idref</p>	
idRef	A	NM/TM	1	<p>Identification of the 'PurchaseData' fragment that SHALL BE purchased before this purchaseData is purchased.</p>	anyURI
PrivateExt	E1	NO/TO	0..1	<p>An element serving as a container for proprietary or application-specific extensions.</p>	
<proprietary elements>	E2	NO/TO	0..N	<p>Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.</p>	

### 5.1.2.8 Purchase Channel

The 'PurchaseChannel' fragment represents a system from which access and content rights can be purchased by the terminal (and its end-user).

Name	Type	Category	Cardinality	Description	Data Type
PurchaseChannel	E			<p>'PurchaseChannel' fragment</p> <p>Contains the following attributes: id version validFrom validTo globalPurchaseChannelID rightsIssuerURI</p> <p>Contains the following elements: PortalURL PurchaseURL Name Description ContactInfo Extension TermsOfUse PrivateExt</p>	
id	A	NM/TM	1	<p>ID of the 'PurchaseChannel' fragment. The value of this attribute SHALL be globally unique.</p>	anyURI

version	A	NM/ TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the 'validFrom' attribute, or as soon as it has been received if no 'validFrom' attribute is given.	unsignedInt
validFrom	A	NM/ TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
globalPurchaseChannelID	A	NM/TM	1	The globally unique identifier of the purchase channel described by this fragment. The 'globalPurchaseChannelID' is referenced by a Coupon document during purchase or other purchase item related actions (see [BCAST12-Services]).	anyURI
rightsIssuerURI	A	NO/ TO	0..1	ID of the rights issuer associated with the BSM. This information is needed to allow unconnected devices to identify the Rights Issuer Services (specified in section 12 of [DRM20-Broadcast-Extensions]) that may be operated by their Home BSM.  The network and terminal SHALL support this attribute when the DRM Profile with broadcast only mode [DRM20-Broadcast-Extensions] is supported.	anyURI
PortalURL	E1	NM/ TM	0..N	The URL on which the BSM offers service related information and/or web-based service provisioning via HTTP or HTTPS. Contains the following attributes: supportedService kmsType At most one PortalURL per associated Key Management System (signalled by the "kmsType" element) SHALL be given. If the network expects the terminal not to use the Web portal for the service provisioning functionality, at most one PortalURL SHALL be instantiated here, with the kmsType attribute absent.	anyURI

supportedService	A	NM/ TM	0..1	<p>Specifies how the Terminal is expected to use 'PortalURL' and 'PurchaseURL'.</p> <p>0: The Terminal SHALL use the Service provisioning messages of [BCAST12-Services] to 'PurchaseURL' to enable the service provisioning functionality. Further, the Terminal MAY contact the 'PortalURL' via HTTP to acquire further information on the purchase items available on this purchase channel.</p> <p>1: The Terminal SHALL be able to access the 'PortalURL' via HTTP to acquire further information on purchase items available on this purchase channel and to achieve service provisioning functionality. Further, the Terminal SHALL perform the service provisioning directly on the 'PortalURL' and SHALL NOT send the Service provisioning messages of [BCAST12-Services] to 'PurchaseURL' to enable the service provisioning functionality.</p> <p>2: The Terminal SHALL be able to access the 'PortalURL' via HTTP to acquire further information on purchase items available on this purchase channel. Further, the Terminal MAY perform the service provisioning directly on the 'PortalURL' or send the Service provisioning messages of [BCAST12-Services] to 'PurchaseURL' to enable the service provisioning functionality.</p> <p>3-255: Reserved for future use.</p> <p>Default: 0</p>	unsignedByte
kmsType	A	NM/TM	0..1	<p>Identifies the type of Key Management System(s)(KMS). This element MAY only be instantiated when the supportedService attribute value is equal to '1' or '2'.</p> <p>Possible values:</p> <p>0. oma-bcast-drm-pki Indicates OMA BCAST DRM profile (Public Key Infrastructure)</p> <p>1. oma-bcast-gba_u-mbms Indicates BCAST Smartcard profile using GBA_U (Symmetric Key Infrastructure)</p> <p>2. oma-bcast-gba_me-mbms Indicates BCAST Smartcard profile using GBA_ME</p> <p>3. oma-bcast-prov-bcmcs Indicates provisioned 3GPP2 BCMCS SKI</p> <p>4 -127 Reserved for future use</p> <p>128 – 255 Reserved for proprietary use</p>	unsignedByte

PurchaseURL	E1	NM/TM	0..N	<p>The URL to which the BCAST Service Provisioning messages as specified in section 5.1 of [BCAST12-Services] SHALL be addressed. At most one PurchaseURL per Key Management System SHALL be given.</p> <p>Contains the following attributes:</p> <p>kmsType bcastrelease</p>	anyURI
kmsType	A	NM/TM	1	<p>Identifies the type of Key Management System(s)(KMS). Possible values:</p> <p>0. oma-bcast-drm-pki Indicates OMA BCAST DRM profile (Public Key Infrastructure)</p> <p>1. oma-bcast-gba_u-mbms Indicates BCAST Smartcard profile using GBA_U (Symmetric Key Infrastructure)</p> <p>2. oma-bcast-gba_me-mbms Indicates BCAST Smartcard profile using GBA_ME</p> <p>3. oma-bcast-prov-bcmcs Indicates provisioned 3GPP2 BCMCS SKI</p> <p>4. KMS not applicable Indicates that no KMS is used. This value SHALL only be used for unencrypted unicast services specified in section 5.10 of [BCAST12-Services].</p> <p>5 -127 Reserved for future use 128 – 255 Reserved for proprietary use</p>	unsignedByte
bcastrelease	A	NM/TM	0..1	<p>Declares the BCAST release according to which the BSP-M in BSM processes Service Provisioning messages from the terminal.</p> <p>Possible values:</p> <p>0. BCAST release 1.0 1. BCAST release 1.1 2. BCAST release 1.2 3 – 255 Reserved for future use</p> <p>Default: 0</p> <p>Note: In case the BSP-M in BSM supports BCAST release 1.1 this attribute SHALL be instantiated and set to '1'. Consequently the terminal SHALL conclude from the absence of this attribute that it is communication with the BSM implemented according to the BCAST release 1.0 specifications. In that case the terminal SHALL use the Service Provisioning messages as specified in the BCAST 1.0 specifications.</p> <p>Note: Even in the case when this attribute is assigned with value '1', the BSM SHALL expect BCAST release 1.0 terminals, if they exist, to perform BCAST release 1.0 based Service Provisioning procedures on this PurchaseURL</p>	unsignedByte

<b>Start of program guide</b>					
				<p>The program guide elements of this fragment are grouped between the Start of program guide and end of program guide cells in this fragment.</p> <p>The program guide elements are for user interpretation. This enables the content creator to provide user readable information about the service. The terminal SHOULD use all declared program guide elements in this fragment for presentation to the end-user. The terminal MAY offer search, sort etc functionalities.</p> <p>The Program Guide consists of the following elements:</p> <p>Name Description ContactInfo Extension</p>	
Name	E1	NM/ TM	1..N	Name of the Purchase Channel, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
Description	E1	NM/ TM	0..N	Description of the purchase channel, possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.	string
ContactInfo	E1	NM/ TM	0..1	A text string that indicates to a user how to contact a BSM to initiate an out-of-band purchase transaction (e.g. phone number, URL etc)	string
Extension	E1	NM/ TM	0..N	Additional information related to this fragment. Contains the following attribute: url Contains the following element: Description	
url	A	NM/ TM	1	URL containing additional information related to this fragment.	anyURI
Description	E2	NM/TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute 'xml:lang' with this element	string
<b>End of program guide</b>					

TermsOfUse	E1	NO/ TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>When Terms of Use is declared for a 'Purchase Channel' fragment the declared Terms of Use SHALL be interpreted to apply all fragments directly linked or indirectly linked (linked via another fragment) with this fragment. Each of those other fragments MAY declare its own Terms of Use which SHALL take precedence of the Terms of Use declared in the 'Purchase Channel' fragment.</p> <p>Contains the following attributes: type id userConsentRequired</p> <p>Contains the following elements: Country Language PreviewDataIDRef TermsOfUseText</p>	
type	A	NM/ TM	1	<p>The way the terminal SHALL interpret the Terms of Use:</p> <p>0 – Display before purchasing or subscribing.</p> <p>If 'TermsOfUse' element of type '0' is present, terminal SHALL render the Terms of Use prior to initiating purchase or subscription request related PurchaseItem associated with this fragment.</p> <p>1 – Not used.</p> <p>2 - 127 reserved for future use</p> <p>128 -255 reserved for proprietary use</p>	unsignedByte
id	A	NM/ TM	1	The URI uniquely identifying the Terms of Use.	anyURI



userConsent Required	A	NM/ TM	1	Signals whether user consent for these Terms of Use is needed. true: User consent is required for these Terms of Use and needs to be confirmed in the subscription / purchase request message related to the PurchaseItem associated with this fragment. false: User consent is not required for the Terms of Use.	boolean
Country	E2	NM/ TM	0..N	List of countries for which the Terms of Use are applicable if consuming the service in that country. Each value is a Mobile Country Code according to [ITU-MCC] If this element is omitted, the Terms of Use are applicable to any country.	string
Language	E2	NM/ TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/ TM	0..1	Reference to the 'PreviewData' fragment which carries the representation of Terms of Use. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/ TM	0..1	Terms of Use text to be rendered. If this element is not present, the 'PreviewDataIDRef' element SHALL be present (Implementation in XML schema using <choice>).	string
PrivateExt	E1	NO/ TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.1.2.9 PreviewData

'PreviewData' fragment contains information about the preview data that is used by the terminal to present the service or content outline to users, so that the users can have a general idea of what the service is about. It can also be used to present an alternative to a blacked-out service. The 'PreviewData' fragment can include simple texts, static binary images (for example, logo), references to images, short video clips and audio clips, , simple rich media content, or even reference to another service which could be a low bit rate version for the main service. Other fragments like 'Service', 'Content', 'Schedule', 'PurchaseData' and 'Access' can link to 'PreviewData' fragment.

Typically the preview data can be used for service-by-service switching, Service Guide browsing, service preview, barker, carrying representation of Terms of Use, alternative in case of blackout, etc. How to use the preview data in different circumstances are further specified in section 5.7.

The network MAY deliver the preview data over broadcast channel using BCAST File Distribution functionality or BCAST Stream Distribution Functionality. Alternatively, the network MAY deliver the preview data over interaction channel.

The terminal SHALL support reception of preview data over broadcast channel. The distribution session for preview data distribution is specified by the 'Access' fragment which is referenced by AccessReference within 'PreviewData' fragment or AccessReference within 'Video', 'Audio' and 'Picture" elements..

If the terminal supports interactive channel, the terminal SHALL support the reception of preview data over interaction channel.

In case the media components of the preview data are delivered via a file distribution session, the preview data SHALL be distributed by the Network before time of distribution of the associated main service, so that the terminal can retrieve, store and show the preview data before the main service is available

Name	Type	Category	Cardinality	Description	Data Type
PreviewData	E			<p>‘PreviewData’ fragment.</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>id</li> <li>version</li> <li>validFrom</li> <li>validTo</li> </ul> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>SMIL</li> <li>Video</li> <li>Audio</li> <li>Picture</li> <li>Text</li> <li>AccessReference</li> <li>RichMedia</li> <li>PrivateExt</li> </ul> <p>In the description below, "SMIL", "Video", "Audio", "Picture", "Text" and "RichMedia" E1 elements are referred to as the "media elements" of the PreviewData fragment.</p>	
id	A	NM/TM	1	<p>ID of the ‘PreviewData’ fragment. The value of this attribute SHALL be globally unique.</p> <p>This ID can be used by other fragments so as to link different service descriptions, purchase information, etc. to the PreviewData.</p>	anyURI
version	A	NM/TM	1	<p>Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no ‘validFrom’ attribute is given.</p>	unsignedInt
validFrom	A	NM/TM	0..1	<p>The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp.</p>	unsignedInt
validTo	A	NM/TM	0..1	<p>The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp.</p>	unsignedInt

SMIL	E1	NM/ TM	0..1	<p>Embedded SMIL to define and synchronize the multimedia components (video, audio, picture, text) of the preview data. This SMIL instance SHALL be well-formed.</p> <p>The content of the 'SMIL' element SHALL either be embedded in a CDATA section or base64-encoded.</p> <p>SMIL (Synchronized Multimedia Integration Language) is a language that allows authors to be able to easily define and synchronize multimedia elements (video, sound, still images) for Web-like presentation and interaction.</p> <p>Contains the following attributes:</p> <p>relativePreference type encoding</p>	string
relativePreference	A	NM/TM	0..1	<p>This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1).</p> <p>If multiple media elements are instantiated in this 'PreviewData' fragment, then all the elements SHALL have the 'relativePreference' attribute instantiated and SHALL have mutually exclusive values of this attribute.</p> <p>If only a single element is instantiated in this 'PreviewData' fragment then the 'relativePreference' attribute MAY be instantiated for that element.</p>	unsignedInt
type	A	NM/ TM	0..1	<p>The type of SMIL profile associated with this PreviewData.</p> <p>0: 3GPP PSS SMIL Profile [3GPP TS 26.246] 1: 3GPP2 MSS SMIL Profile [3GPP2 C.S0050] 2 - 127: reserved for future use 128 - 255: reserved for proprietary use</p> <p>If this attribute is not present, the SMIL profile is unspecified.</p>	unsignedByte
encoding	A	NM/TM	0..1	<p>This attribute signals the way the SMIL data have been embedded:</p> <ul style="list-style-type: none"> <li>• It SHALL NOT be present when the SMIL data are embedded into a CDATA section.</li> <li>• It SHALL be present and set to "base64" in case the SMIL data are base64-encoded.</li> </ul>	string

Video	E1	NM/ TM	0..1	<p>Video defines how to obtain an audio/video trailer clip which can enable the user to preview the service or content.</p> <p>Contains the following attribute: relativePreference</p> <p>Contains the following elements: VideoURI MIMEType AlternativeText AlternativePicture AccessReference</p>	
relativePreference	A	NM/TM	0..1	<p>This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1).</p> <p>If multiple media elements are instantiated in this 'PreviewData' fragment, then all the elements SHALL have the 'relativePreference' attribute instantiated and SHALL have mutually exclusive values of this attribute.</p> <p>If only a single element is instantiated in this 'PreviewData' fragment then the 'relativePreference' attribute MAY be instantiated for that element.</p>	unsignedInt
VideoURI	E2	NM/ TM	1	<p>The URI referencing the video clip.</p> <p>When ALC is used for delivery of the video clip, this corresponds to the 'Content-Location' attribute' in the File element in the 'Access' fragment.</p> <p>When FLUTE is used for delivery of the video clip, this corresponds to the 'Content-Location' attribute in the FDT of the FLUTE session.</p> <p>When HTTP is used for delivery of the video clip, the rules defined for usage of HTTP with the 'contentLocation' attribute of the 'Schedule' fragment apply (see section 5.1.2.2 of this specification).</p> <p>When RTSP is used for negotiation of the video clip delivery, the rules defined for usage of RTSP with the 'contentLocation' attribute of the 'Schedule' fragment apply (see section 5.1.2.2 of this specification).</p>	anyURI
MIMEType	E2	NM/ TM	0..1	<p>MIME Media type of the video clip.</p> <p>Contains the following attribute: codec</p>	string

codec	A	NO/ TM	0..1	The codec parameters for the associated MIME Media type. If the file's MIME type definition specifies mandatory parameters, these MUST be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the file SHOULD be included in the string. One example of the parameters defined for video/3GPP, video/3GPP2 is specified in [RFC4281].	string
AlternativeText	E2	NM/ TM	0..N	Alternative Text to be displayed if the video clip is not available. Possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element. The same schema of element 'Text' is used for 'AlternativeText'.	
AlternativePicture	E2	NO/ TM	0..1	Alternative Picture to be displayed if the video clip is not available. AlternativePicture can be PictureData or URI reference of the Picture. The same schema of element 'Picture' is used for 'AlternativePicture'	
AccessReference	E2	NM/TM	0..1	ID of the 'Access' fragment that specifies the delivery of the video file. If this element is instantiated and 'AccessReference' E1 Element in 'previewData' fragment is also instantiated, the value of this element overrides that of E1 'AccessReference'. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Access' fragment with which the 'Video' element is associated.	anyURI
Audio	E1	NM/ TM	0..1	Audio defines how to obtain an audio clip which can enable the user to preview the service or content.  Contains the following attribute: relativePreference Contains the following elements: AudioURI MIMEType AlternativeText AlternativePicture AccessReference	

relativePreference	A	NM/TM	0..1	<p>This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1).</p> <p>If multiple media elements are instantiated in this 'PreviewData' fragment, then all the elements SHALL have the 'relativePreference' attribute instantiated and SHALL have mutually exclusive values of this attribute.</p> <p>If only a single element is instantiated in this 'PreviewData' fragment then the 'relativePreference' attribute MAY be instantiated for that element.</p>	unsignedInt
AudioURI	E2	NM/TM	1	<p>The URI referencing the audio clip.</p> <p>When ALC is used for delivery of the audio clip, this corresponds to the 'Content-Location' attribute' in the File element in the 'Access' fragment.</p> <p>When FLUTE is used for delivery of the audio clip, this corresponds to the 'Content-Location' attribute in the FDT of the FLUTE session.</p> <p>When HTTP is used for delivery of the audio clip, the rules defined for usage of HTTP with the 'contentLocation' attribute of the 'Schedule' fragment apply (see section 5.1.2.2 of this specification).</p> <p>When RTSP is used for negotiation of the audio clip delivery, the rules defined for usage of RTSP with the 'contentLocation' attribute of the 'Schedule' fragment apply (see section 5.1.2.2 of this specification).</p>	anyURI
MIMEType	E2	NM/TM	0..1	<p>MIME Media type of the audio clip.</p> <p>Contains the following attribute: codec</p>	string
codec	A	NO/TM	0..1	<p>The codec parameters for the associated MIME Media type. If the file's MIME type definition specifies mandatory parameters, these MUST be included in this string. Optional parameters containing information that can be used to determine as to whether the terminal can make use of the file SHOULD be included in the string. One example of the parameters defined for audio/3GPP, audio/3GPP2 is specified in [RFC4281].</p>	string
AlternativeText	E2	NM/TM	0..N	<p>Alternative Text to be displayed if the audio clip is not available. Possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element.</p> <p>The same schema of element 'Text' is used for 'AlternativeText'.</p>	
AlternativePicture	E2	NO/TM	0..1	<p>Alternative Picture to be displayed if the audio clip is not available. AlternativePicture can be PictureData or URI reference of the Picture.</p> <p>The same schema of element 'Picture' is used for 'AlternativePicture'.</p>	

AccessReference	E2	NM/TM	0..1	ID of the 'Access' fragment that specifies the delivery of the audio file. If this element is instantiated and 'AccessReference' E1 Element in 'previewData' fragment is also instantiated, the value of this element overrides that of E1 'AccessReference'. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Access' fragment with which the 'Audio' element is associated.	anyURI
Picture	E1	NM/TM	0..1	Picture defines how to obtain a picture which can enable the user to preview the service or content, or represents the service or content with an icon, a logo, etc. The associated picture can be represented with binary data embedded in the 'PreviewData' fragment or be referenced by URI  Contains the following attribute: relativePreference  Contains the following elements: PictureURI PictureData MIMEType AlternativeText	
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media elements are instantiated in this 'PreviewData' fragment, then all the elements SHALL have the 'relativePreference' attribute instantiated and SHALL have mutually exclusive values of this attribute. If only a single element is instantiated in this 'PreviewData' fragment then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt
PictureURI	E2	NM/TM	0..1	The URI referencing the picture. When ALC is used for delivery of the picture file, this corresponds to the 'Content-Location' attribute' in the File element in the 'Access' fragment. When FLUTE is used for delivery of the picture file, this corresponds to the 'Content-Location' attribute in the FDT of the FLUTE session. When HTTP is used for delivery of the picture file, the rules defined for usage of HTTP with the 'contentLocation' attribute of the 'Schedule' fragment apply (see section 5.1.2.2 of this specification). Either PictureURI or PictureData MUST be used if Picture element is present.	anyURI

PictureData	E2	NM/ TM	0..1	The binary data of the picture, encoded in Base64 to allow embedding into XML. Either PictureURI or PictureData MUST be used if Picture element is present.	base64Binary
MIMEType	E2	NM/ TM	0..1	MIME Media type of the picture.	string
AlternativeText	E2	NM/ TM	0..N	Alternative Text to be displayed if the picture is not available. Possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element. The same schema of element 'Text' is used for 'AlternativeText'.	
AccessReference	E2	NM/TM	0..1	ID of the 'Access' fragment that specifies the delivery of the picture file. If this element is instantiated and 'AccessReference' E1 Element in 'previewData' fragment is also instantiated, the value of this element overrides that of E1 'AccessReference'. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Access' fragment with which the 'Picture' element is associated.	anyURI
Text	E1	NM/ TM	0..N	Text represents textual preview of the service or content. For example, the contents of this element can be a title sentence, or brief description of service or content for preview purposes. Possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element. Text format attributes (e.g. font, size and colour) are defined by HTML version 4.01. The content of the 'Text' element SHALL either be embedded in a CDATA section or base64-encoded. This element SHALL NOT override the value of element 'Name' in 'Service' or 'Content' fragment. Contains the following attribute: relativePreference encoding	string
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media elements are instantiated in this 'PreviewData' fragment, then all the elements SHALL have the 'relativePreference' attribute instantiated and SHALL have mutually exclusive values of this attribute. If only a single element is instantiated in this 'PreviewData' fragment then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt



encoding	A	NM/TM	0..1	This attribute signals the way the HTML data have been embedded: <ul style="list-style-type: none"> <li>It SHALL NOT be present when the HTML data are embedded into a CDATA section.</li> <li>It SHALL be present and set to “base64” in case the HTML data are base64-encoded.</li> </ul>	string
AccessReference	E1	NM/TM	0..1	ID of the ‘Access’ fragment that specifies the delivery of the associated preview data. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the ‘Access’ fragment which this ‘PreviewData’ fragment is associated with.	anyURI
RichMedia	E1	NO/TO	0..N	RichMedia defines how to obtain the rich media content which can enable the user to preview the service or content. Contains the following attribute: relativePreference Contains the following elements: Capabilities RichMediaData RichMediaURI AlternativeText AlternativePicture	
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this element. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media elements are instantiated in this ‘PreviewData’ fragment, then all the elements SHALL have the ‘relativePreference’ attribute instantiated and SHALL have mutually exclusive values of this attribute. If only a single element is instantiated in this ‘PreviewData’ fragment then the ‘relativePreference’ attribute MAY be instantiated for that element.	unsignedInt
Capabilities	E2	NM/TM	1	Describes the type and complexity of Rich Media Solution the rich media engine has to deal with.	complexType as defined in section 5.1.2.4 for Capabilities element child of RichMedia element in Access fragment

RichMediaData	E2	NM/TM	0..1	An inlined Rich Media content that SHALL either be embedded in a CDATA section or base64-encoded. Contains the following attribute: encoding  Either RichMediaURI or RichMediaData MUST be used if RichMedia element is present.	
encoding	A	NM/TM	0..1	This attribute signals the way the rich media data have been embedded: <ul style="list-style-type: none"> <li>It SHALL NOT be present when the rich media data are embedded into a CDATA section.</li> </ul> Note: binary data inside CDATA shall always be encoded in base64 <ul style="list-style-type: none"> <li>It SHALL be present and set to “base64” in case the rich media data are base64-encoded</li> </ul>	String
RichMediaURI	E2	NM/TM	0..1	The URI referencing the rich media content.  When ALC is used for delivery of the rich media content, this corresponds to the ‘Content-Location’ attribute’ in the File element in the ‘Access’ fragment.  When FLUTE is used for delivery of the rich media content, this corresponds to the ‘Content-Location’ attribute in the FDT of the FLUTE session.  When HTTP is used for delivery of the rich-media content, the rules defined for usage of HTTP with the ‘contentLocation’ attribute of the ‘Schedule’ fragment apply (see section 5.1.2.2 of this specification).  When RTSP is used for negotiation of the rich media content delivery, this corresponds to the ‘Request-URI’ to be used in the request line of RTSP request.	anyURI
AlternativeText	E2	NM/TM	0..N	Alternative Text to be displayed if the rich media content is not available. Possibly in multiple languages. The language is expressed using built-in XML attribute ‘xml:lang’ with this element.  The same schema of element ‘Text’ is used for ‘AlternativeText’.	

AlternativePicture	E2	NO/TM	0..1	Alternative Picture to be displayed if the rich media content is not available. AlternativePicture can be PictureData or URI reference of the Picture.  The same schema of element 'Picture' is used for 'AlternativePicture'.	
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.1.2.10 InteractivityData

The 'InteractivityData' fragment is used to associate services and/or individual pieces of content of the services with interactivity components of service/content consumption. These interactivity components are used by the terminal to offer interactive services to the user possibly in parallel with the 'regular' broadcast content. These interactivity services enable users to e.g. vote during TV shows or to obtain content related to the 'regular' broadcast content. Whereas the 'InteractivityData' fragment can be thought to declare the availability of the interactivity components, the details of the components are provided via one or many InteractivityMediaDocuments (see [BCAST12-Services] section 5.3.6.1) that may include xhtml files, static images, email template, SMS template, MMS template documents, etc.

The 'InteractivityData' fragment SHALL always refer to one 'Service' fragment. The 'InteractivityData' fragment can also reference 'Content' fragment, 'Schedule' fragment or Interactivity Windows to further narrow down the association by linking the interactivity with part of the service. The presence of 'ContentReference' element and 'ScheduleReference' element in the 'InteractivityData' fragment SHALL be mutually exclusive.

In all cases, the time interval when the media object sets of an InteractivityMediaDocument are valid to be rendered, on the terminal, is declared by 'validFrom' and 'validTo' attributes in the InteractivityMediaDocument. Whenever there are more than one 'InteractivityData' fragments valid at the same time for a certain service, the following priority SHALL be maintained by the terminal for processing and rendering the associated interactivity. However how the terminal actually processes and renders the simultaneous interactivities with different priorities is out of the scope of this specification. The time interval during which the interactivity can be activated (e.g., user is able to give a vote) in different cases is also indicated.

- Priority 1 (Highest) – 'ServiceReference' element and 'InteractivityWindow' element are present in the InteractivityData' fragment:

In this case the interactivity is associated with the declared interactivity windows of this service and the interactivity can be activated during those interactivity windows.

- Priority 2 – 'ServiceReference' element, 'ScheduleReference' element and 'ScheduleReference.PresentationWindowIDRef' sub-element are present:

In this case the interactivity is associated with the indicated subset of the presentation windows of the service which are originally declared in the referenced Schedule fragment, and this interactivity can be activated during these subset of presentation windows.

- Priority 3 – 'ServiceReference' element and 'ScheduleReference' element are present, but 'ScheduleReference.PresentationWindowIDRef' sub-element is not present:

In this case the interactivity is associated with all presentation windows of this service which are originally declared in the referenced Schedule fragment, and this interactivity can be activated during all these presentation windows.

- Priority 4 – 'ServiceReference' element and 'ContentReference' element are present

In this case the interactivity is associated with the content of this service and can be activated during the life span of the content.

- Priority 5 (Lowest) – only ‘ServiceReference’ element is present

In this case the interactivity is associated with this service and can be activated during the life span of the service.

Terminals with no return channel MAY NOT support ‘InteractivityData’ fragments. Terminals with a return channel (e.g. UMTS, GPRS) SHALL support ‘InteractivityData’ fragments.

The concept of grouping of InteractivityMediaDocuments (as defined in [BCAST12-Services] section 5.3.6.1) allows

- to group multiple InteractivityMediaDocuments into one logical unit representing the same interactivity,
- but also to associate InteractivityMediaDocuments with ‘InteractivityData’ fragments.

The network SHALL assign each ‘InteractivityData’ fragment one and only one InteractivityMediaDocument group identifier (InteractivityMediaDocumentPointer) which is used by the terminals to identify the InteractivityMediaDocument belonging to the interactivity represented by the ‘InteractivityData’ fragment in question.

The media objects listed in the InteractivityMediaDocuments associated with the ‘InteractivityData’ fragment can be distributed before or at the same time of distribution of the ‘regular’ broadcast media stream. An InteractivityMediaDocument distributed before that time can be cached by the terminal. This is indicated by setting the Pre-listenIndicator to “true”.

InteractivityMediaDocuments can be distributed over the same access channel as the service they are associated with, or over a different access channel. Distribution over a different access is enabled by association of an ‘InteractivityData’ fragment to a ‘Schedule’ fragment that is referred to by a different ‘Access’ fragment than service.

In case multiple groups of InteractivityMediaDocuments are transported using the same file delivery session, the network MUST assign mutually exclusive InteractivityMediaDocument group identifiers to the ‘InteractivityData’ fragments in question. This is in order to enable the terminals to distinguish between the InteractivityMediaDocuments of the different interactivities.

‘InteractivityData’ fragment can specify that interaction sent back from device to service provider shall be distributed over time, e.g. to avoid overload in network nodes or links caused by too many simultaneous interactivity messages sent back. This is done by declaring a time window during which interactivity shall be sent back. The declaration of this information in an ‘InteractivityData’ fragment applies to all interactivity declared through this ‘InteractivityData’ fragment, but can be superseded by explicit declaration of the same information per interaction, in an InteractivityMediaDocument.

If interaction is announced using ‘InteractivityData’ fragment, the service provider SHOULD deliver InteractivityMediaDocuments corresponding to that ‘InteractivityData’ fragment.

Name	Type	Category	Cardinality	Description	Data Type
------	------	----------	-------------	-------------	-----------

Interactivity Data	E			<p>‘InteractivityData’ fragment. Contains the following attributes: id version validFrom validTo preListenIndicator interactivityMediaDocumentPointer Contains the following elements: InteractivityType ServiceReference ContentReference ScheduleReference InteractivityWindow InteractiveDelivery Extension BackOffTiming TermsOfUse TargetUserProfile PrivateExt</p>	
id	A	NM/TM	1	ID of the ‘InteractivityData’ fragment. The value of this attribute SHALL be globally unique.	anyURI
version	A	NM/TM	1	Version of this fragment. The newer version overrides the older one starting from the time specified by the ‘validFrom’ attribute, or as soon as it has been received if no ‘validFrom’ attribute is given.	unsignedInt
validFrom	A	NM/TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp	unsignedInt
validTo	A	NM/TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp	unsignedInt
preListenIndicator	A	NM/TM	1	<p>If the attribute ‘prelistenIndicator’ is “true” the terminal SHOULD retrieve and locally store the Interactivity media objects included in the InteractivityMediaDocuments carried in the broadcast stream (see [BCAST12-Services] section 5.3.6). The terminal SHOULD start the retrieval of these Interactivity media objects prior to the broadcast time of the ‘Service’, ‘Content’, ‘Schedule’ or ‘InteractivityWindow’ it is associated with or as soon as the ‘InteractivityData’ fragment is retrieved by the terminal.</p> <p>If the attribute ‘prelistenIndicator’ is “false” the terminal MAY retrieve the Interactivity media objects included in the InteractivityMediaDocuments, before the ‘Service’, ‘Content’, ‘Schedule’ or ‘InteractivityWindow’ it is associated with, is broadcasted.</p>	boolean

interactivityMediaDocumentPointer	A	NM/TM	1	Reference to the GroupID of the InteractivityMediaDocuments which refer to the interactivity media objects. The pointer points to all InteractivityMediaDocuments with the same GroupID. If TargetUserProfile is present and matches user preference, then the InteractivityMediaDocument which has the highest GroupPosition (see [BCAST12-Services] section 5.3.6) is rendered; else, the InteractivityMediaDocument with the highest GroupPosition is rendered. When multiple 'InteractivityData' fragments point to the same GroupID, this means that the InteractivityMediaDocuments belonging to this GroupID applies to all of these 'InteractivityData' fragments.	anyURI
InteractivityType	E1	NO/TO	0..N	Description of the type of the Interactive Service, such as "voting" or "gambling", possibly in multiple languages. The language is expressed using built-in XML attribute 'xml:lang' with this element. Terminal devices supporting Interactive channel SHALL support this attribute. It SHALL be used for rendering only.	string
ServiceReference	E1	NM/TM	1	Reference to the 'Service' fragment that the 'InteractivityData' fragment is associated with. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Service' fragment which this 'InteractivityData' fragment is associated with.	anyURI
ContentReference	E1	NM/TM	0..N	Reference to the 'Content' fragments that the 'InteractivityData' fragment is associated with. If this element is present, that means the interactivity is associated with the referenced content. If there is no 'InteractivityWindow' element declared in this 'InteractivityData' fragment the terminal SHOULD assume the interactivity can be activated during the life span of the content, i.e. the time the content is accessed or rendered. The actual validity for the terminal to access the interactivity is declared by 'validFrom' and 'validTo' attributes in the corresponding InteractivityMediaDocuments. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Identification of the 'Content' fragment which this 'InteractivityData' fragment is associated with.	anyURI

ScheduleReference	E1	NM/TM	0..N	<p>Reference to the ‘Schedule’ fragments that the ‘InteractivityData’ fragment is associated with.</p> <p>If this element is present, that means the interactivity is associated with the referenced schedule. If there is no 'InteractivityWindow' element declared in this 'InteractivityData' fragment the terminal SHOULD assume the interactivity can be activated during the lifespan of the schedule, i.e. the time the content is accessed or rendered through the schedule or a subset of the presentation windows of the schedule indicated by ‘PresentationWindowIDRef’ sub-element.</p> <p>The actual validity for the terminal to access the interactivity is declared by ‘validFrom’ and ‘validTo’ attributes in the corresponding InteractivityMediaDocuments.</p> <p>Contains the following attribute: idRef</p> <p>Contains the following element: PresentationWindowIDRef</p>	
idRef	A	NM/TM	1	The id of the ‘Schedule’ fragment this ‘InteractivityData’ fragment refers to, globally unique.	anyURI
PresentationWindowIDRef	E2	NM/TM	0..N	<p>Relation reference to the PresentationWindow to which the ‘InteractivityData’ fragment belongs.</p> <p>The ‘PresentationWindowIDRef’ declared in this sub-element SHALL be the complete collection or a subset of the PresentationWindow ids declared in the ‘Schedule’ fragment referenced by the above Schedule ‘idRef’. If the latter case is true, then the interactivity SHALL only be assumed to be associated with these presentation windows, as opposed to during every presentation windows declared in the referenced ‘Schedule’ fragment.</p> <p>If ‘PresentationWindowIDRef’ element is absent, the interactivity SHALL be assumed to be associated with every presentation windows declared in the referenced ‘Schedule’ fragment.</p>	unsignedInt

Interactivity Window	E1	NM/TM	0..N	Time interval during which this 'InteractivityData' fragment is associated with the service specified by 'ServiceReference' element. If this element is present, that means the interactivity is associated with the declared InteractivityWindows and terminal SHOULD assume the interactivity can be activated during any of those InteractivityWindows. The actual validity for the terminal to access the interactivity is declared by 'validFrom' and 'validTo' attributes in the corresponding InteractivityMediaDocuments. Contains the following attributes: startTime endTime	
startTime	A	NM/TM	1	Start of the InteractivityWindow. Whenever an InteractivityWindow is specified, StartTime SHALL be declared. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
endTime	A	NM/TM	1	End of the InteractivityWindow. Whenever an InteractivityWindow is specified, EndTime SHALL be declared. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
InteractiveDelivery	E1	NO/TM	0..1	This element indicates the possibility to receive InteractivityMedia over the interaction channel. Interactivity Media can either be pushed, using OMA PUSH delivery, or pulled, using HTTP requests to InteractivityMediaURL. If this element is present, at least one of PushDelivery and InteractivityMediaURL shall be included. Contains the following attributes: interactivityMediaURL pushDelivery	
interactivityMediaURL	A	NO/TM	0..1	URL from which Interactivity Media can be retrieved. The Content-Type SHALL be "multipart/mixed" in the HTTP response.	anyURI
pushDelivery	A	NO/TM	0..1	If this attribute is present and "true", the terminal SHALL expect the delivery of InteractivityMedia using OMA PUSH, as described in [BCAST12-Distribution] section 9. In this case, the PUSH messages contain InteractivityMedia whose content type is "application/vnd.oma.bcast.imd+xml". The default of this attribute is "false".	boolean
Extension	E1	NM/TM	0..1	Additional information related to this fragment. Contains the following attribute: url Contains the following element: Description	
url	A	NM/TM	1	URL containing additional information related to this fragment.	anyURI



Description	E2	NM/TM	0..N	Description regarding the additional information which can be retrieved from a web page. The language is expressed using built-in XML attribute xml:lang with this element	string
BackOffTiming	E1	NM/TM	0..1	<p>This element specifies default timing behaviour of interaction sent back from the device to the service provider. Its purpose is to provide a mechanisms that ensures distribution over time of feedback sent from receivers, e.g. in order to avoid overload in nodes or links.</p> <p>If present, the interaction, if any, SHALL be sent back in the time interval [offsetTime, offsetTime+randomTime] after the event that triggers the interactivity (e.g. user feedback). The exact time within the allowed time window shall be random with uniform probability.</p> <p>The declaration of this information in InteractivityData applies as a default to all interactivity declared through this InteractivityData, but if explicit timing behaviour is expressed in InteractivityMediaDocument, it prevails over default timing behaviour expressed in 'InteractivityData' fragment.</p> <p>Contains the following attributes: offsetTime randomTime</p>	
offsetTime	A	NM/TM	1	The 'offsetTime' attribute specifies the minimum time that a device SHALL wait after an event that triggers interaction (e.g. user input), before sending the interaction. The unit is seconds (fractions can be expressed using data type Decimal). 'offsetTime' shall be a non-negative number.	decimal
randomTime	A	NM/TM	1	<p>The 'randomTime' refers to the time window length over which a device SHALL calculate a random time for the transmission of interaction. The method provides for statistically uniform distribution over a relevant period of time.</p> <p>The device SHALL calculate a uniformly distributed random time out of the interval between 0 and randomTime. The unit is seconds (fractions can be expressed using data type Decimal). 'randomTime' shall be a non-negative number.</p>	decimal

TermsOfUse	E1	NO/TO	0..N	<p>Element that declares there are Terms of Use associated with this fragment.</p> <p>Contains the textual presentation of Terms of Use or a reference to Terms of Use representation through 'PreviewData', and information whether user consent is required for the Terms of Use.</p> <p>Multiple occurrences of 'TermsOfUse' are allowed within this fragment, but for any two such occurrences values for elements 'Country' and 'Language' SHALL NOT be same at the same time.</p> <p>In addition to Terms of Use this element MAY be used for disclaimers, legal text and other pieces of information to be rendered to the user upon activation, purchase or consumption of service or content.</p> <p>Contains the following attributes:</p> <p>type id userConsentRequired</p> <p>Contains the following elements:</p> <p>Country Language PreviewDataIDRef TermsOfUseText</p>	
type	A	NM/TM	1	<p>The way the terminal SHALL interpret the Terms of Use:</p> <p>0 – Not used.</p> <p>1 – Display before payout.</p> <p>If 'TermsOfUse' element of type '1' is present, terminal SHALL present the Terms of Use prior to playing out content or service associated with this fragment.</p> <p>2 - 127 reserved for future use 128 -255 reserved for proprietary use</p>	unsignedByte
id	A	NM/TM	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsentRequired	A	NM/TM	1	<p>Signals whether user consent for these Terms of Use is needed.</p> <p>true: User consent is required for these Terms of Use and needs to be confirmed. How such confirmation is done is out of scope of this specification.</p> <p>false: User consent is not required for the Terms of Use.</p>	boolean
Country	E2	NM/TM	0..N	<p>List of countries for which the Terms of Use are applicable if using the interactivity in that country. Each value is a Mobile Country Code according [ITU-MCC].</p> <p>If this element is omitted, the Terms of Use are applicable to any country.</p>	string of 3 digits

Language	E2	NM/TM	1	Language in which the Terms of Use is given. Value is a three character string according to ISO 639-2 alpha standard for language codes.	string
PreviewData IDRef	E2	NO/TM	0..1	Reference to the 'PreviewData' fragment which carries the representation of Terms of Use. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUse Text	E2	NO/TM	0..1	Terms of Use text to be rendered. If this element is not present the 'PreviewDataIDRef' element SHALL be present (Implementation in XML schema using <choice>).	string
TargetUserProfile	E1	NO/TO	0..N	<p>Profile attributes of the users whom the interaction service is targeting at. The detailed personal attribute names and the corresponding values are specified by attributes of 'attributeName' and 'attributeValue'. Amongst the possible profile attribute names are age, gender, occupation, etc. (subject to national/local rules &amp; regulations, if present and as applicable regarding use of personal profiling information and personal data privacy).</p> <p>The extensible list of 'attributeName' and 'attributeValue' pairs for a particular service enables end user profile filtering and end user preference filtering of broadcast services. The terminal SHOULD be able to support 'TargetUserProfile' element. The terminal behavior for interpreting and acting upon 'TargetUserProfile' is out of the scope.</p> <p>It is RECOMMENDED that use of 'TargetUserProfile' element is an "opt-in" capability for users. Terminal settings SHOULD allow users to configure whether to input their personal profile or preference and whether to allow broadcast service to be automatically filtered based on the users' personal attributes without users' request.</p> <p>Contains the following attributes:  attributeName  attributeValue</p>	
attributeName	A	NM/TM	1	Profile attribute name	string
attributeValue	A	NM/TM	1	Profile attribute value	string
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

## 5.2 Announcing Service Guides within a Service Guide

It is possible to announce other Service Guides within a Service Guide. For the sake of clarity, the Service Guide that refers to another Service Guide is called *referring Service Guide* while the Service Guide that is referred to is called *referred Service Guide*. There are two use cases that announcing other Service Guides within a Service Guide enables:

- **Service Guide complementing another Service Guide:** For example, basic (referring) Service Guide may describe basic services on top level and while the complementing (referred) Service Guides may describe further details of services and/or additional services. In this case the referred Service Guides will need the baseline information in the referring Service Guide.
- **Referring to standalone Service Guide:** For example, basic (referring) Service Guide may aggregate a number of stand-alone (referred) Service Guides by reference. In this case the referred Service Guides will be fully usable even without the referring Service Guide.
- Referring Service Guide MAY include ‘ReferredSGInfo’ in ‘ServiceClass’ element of the ‘Access’ fragment so that additional information related to referred Service Guide can be provided.

The following applies when Service Guides are announced from other Service Guides:

- If the existence of and access to referred Service Guide is announced, the ‘Access’ fragment SHALL be used to announce those:
  - Referring Service Guide MAY include ‘ServiceClass’ element in the ‘Access’ fragment with value urn:oma:bcast:oma\_bsc:csg:1.0 meaning that the referred Service Guide provides complementary information to the referring Service Guide.
  - Referring Service Guide MAY include ‘ServiceClass’ element in the ‘Access’ fragment with value urn:oma:bcast:oma\_bsc:sg:1.0 meaning that the referred Service Guide is a stand-alone Service Guide.
- Regarding distribution of referred and referring Service Guides
  - In case of broadcast delivery of the referred Service Guide, the related ‘Access’ fragment SHALL instantiate the ‘BroadcastDelivery’ element so as to declare a FLUTE delivery session. Such session SHALL transport the SGDD and MAY transport the SGDU related to the referred Service Guide, with their MIME type set as specified in section 5.4.2 of the present document.
  - In case of delivery of the referred Service Guide over the Interaction Channel, the related ‘Access’ fragment SHALL instantiate the ‘UnicastServiceDelivery’ element, with the ‘type’ child attribute set to value “0” (HTTP), and one or more ‘AccessServerURL’ attribute that SHALL be provisioned with the complete URL of the Service Guide server, for the terminal to request the referred Service Guide over the Interaction Channel as specified in section 5.4.3 of the present document.
  - Both delivery mechanisms specified above SHALL solely deliver SGDD (and optionally SGDU) for the referred Service Guide they target.
  - Similarly, the delivery session of a referring Service Guide SHALL NOT carry SGDD and SGDU of any referred Service Guide.
- Regarding consistency of the Service Guides in the complementing case
  - In such case, the Network SHALL ensure that the complete Service Guide, made of the referring and referred Service Guide(s), is consistent (as defined in section 5.4.1.5.3). It is understood that each referred Service Guide(s) can be individually inconsistent. The referring Service Guide SHALL be consistent. The terminal SHALL parse all of the referring and referred Service Guides in order to get a consistent view of the complete Service Guide as declared by the referring part.
- Regarding fragment overriding

- The Service Guide within Service Guide functionality is meant to provide additional fragments to a base Service Guide and as such SHALL NOT be used to override a fragment of the referring Service Guide with a fragment of the referred Service Guide. This implies that fragment identifiers SHALL be uniquely identified within the reference tree.
- As of BCAST 1.1, a referred Service Guide SHALL NOT be a referring Service Guide. The Terminal SHALL support a referencing tree of depth 1 (i.e. it can support one level of reference)
- If all 'Access' fragments associated with a 'Service' fragment announce referred Service Guides, the value of the 'ServiceType' element of 'Service' fragment SHALL be set to "8 – Service Guide".
- If the referred Service Guide is complementary Service Guide, the SG-C SHOULD assume the existence of referring Service Guide to make use of information provided in the referred Service Guide.
- If the referred Service Guide is stand-alone Service Guide, the SG-C SHOULD NOT assume the existence of referring Service Guide to make use of information provided in the referred Service Guide.

The following aspects of Service Guide within Service Guide have been left for further consideration:

- usage in conjunction with the 'GroupingCriteria' feature of the SGDD, in particular when multiple BSMs are declared
- Scope of SGDDs from both referring and referred Service Guides for functions such as Notification and Broadcast Roaming

SG-C SHALL support announcing other Service Guides within a Service Guide and SG-D MAY support announcement of other Service Guides within a Service Guide as defined above.

## 5.3 Interfaces for Service Guide Generation

Referring to OMA BCAST Architecture [BCAST11-Architecture], this section normatively specifies the interfaces SG-1, SG-2, SG-4, SG-5 and SG-6 for Service Guide functionality.

### 5.3.1 Service Guide Data Model as Generic Interchange format

The Service Guide Data Model (as specified in section 5.1) and its XML Schema definition (as specified in [BCAST12-Schema-sg-f]) SHALL be used as generic interchange format for communicating Service Guide entries as fragments over interfaces SG-2, SG-4 and SG-B1 (system back-end interfaces) and over interfaces SG-5 and SG-6 (interfaces between terminal and system). Over interface SG-1 the BCAST XML schema definition MAY be used.

The methods to deliver Service Guide fragments over interfaces SG-1, SG-2 and SG-4 are defined in section 5.4.

The methods to deliver Service Guide fragments over interface SG-B1 are specific to each BDS in question and are covered in the BDS adaptation specifications (see [BCAST12-DVBH-IPDC-Adaptation], [BCAST12-MBMS-Adaptation], [BCAST12-BCMCS-Adaptation], [BCAST12-DVBSH-IPDC-Adaptation], [BCAST12-FLO-Adaptation], [BCAST12-NGH-Adaptation]).

The method to deliver Service Guide fragments over interface SG-5 is specified in section 5.4.2.

The method to deliver Service Guide fragments over interface SG-6 is specified in section 5.4.3.

## 5.4 Service Guide Delivery

Instantiated Service Guide consists of the Service Guide XML fragments and the Session Description information fragments described in the previous sections. Each fragment is designed to be uniquely identifiable in the Service Guide making it possible to treat the Service Guide as a set of these fragments. This naturally makes it possible for the network to divide the set further into subsets each containing only some of the fragments in the set but not necessarily all of them. The network can deliver the Service Guide using these subsets each subset independent of the other instead of placing the full set to the transport as one atomic unit. This arrangement enables the network to isolate the changes on the delivery layer when only

some of the fragments in the Service Guide are altered. In such a case the terminals that have already received the full set of fragments at least once, can utilise the isolation of the changes by receiving only that subset of fragments that contains the changes.

The delivery of the Service Guide can be provided by two means

- over the interactive channel the terminal requesting some or all of the fragments of the Service Guide explicitly or
- over the broadcast channel the terminals polling the fragments.

While both mechanisms have their unique characteristics the mechanisms essentially share the data structures used to manage the organization of the fragments on the transport layer. These data structures are the 'Service Guide Delivery Descriptor' and the 'Service Guide Delivery Unit' short-handed SGDD and SGDU respectively. The latter is the structure that the network uses to encapsulate fragment subsets for the transport layer and it is purely transport independent. The former, the SGDD, on the other hand is partially transport dependent. The SGDD both describes service level information about the Service Guide as well how each of the Service Guide fragment is available for the terminal in the transport layer.

The following sections are organised to address transport layer independent mechanisms and definitions first followed by the transport level dependent sections.

## 5.4.1 Encapsulating and identifying fragments

As discussed before, each of the Service Guide fragments as an XML fragment or Session Description is assigned an identifier for distinguishing any two fragments from each other. These identifiers are called fragment identifiers and they are represented with URIs. This identification scheme naturally makes the fragments not only unique in the scope of a single Service Guide but in fact over all the Service Guides.

In the transport frames, however, one prefers the use of integer based identification schemes over string based identification schemes. This is achieved by assigning for each fragment in addition to the fragment identifier a transport identifier. In the transport layer the fragments are addressed using the transport identifiers the transport layer also providing the binding between these two types of identifiers.

While the specifics of the common data structures SGDD and SGDU are discussed in the later sections, the strong dependency between the two structures imposed by the transport identifiers is defined first.

### 5.4.1.1 Fragment identifier versus transport identifier

As discussed before any two Service Guide fragments can be distinguished by their fragment identifiers. The way these identifiers are represented depends on the type of the fragment:

- In case of XML fragments the identifier can be found in the fragment itself as top level attribute 'id' and
- In case of Session Description the identifier is introduced in the SGDU carrying the fragment using the field 'fragmentID'.

The transport identifiers used in the transport layer to identify fragments are given in the 'fragmentTransportID' fields of the SGDUs carrying the corresponding fragments. The binding between the fragment identifier and the transport identifier is provided using the SGDD where each fragment is declared using an XML element 'Fragment'. This element gives both the fragment identifier and the transport identifier for the fragment providing explicit binding between the two. In order for the binding to be unambiguous the network SHALL ensure the following:

- During the whole lifetime of a Service Guide fragment the 'id' attribute of the fragment is always bound to the same 'fragmentTransportID' value.
- For each Service Guide entry point (see section 6.1.1), the binding between 'fragmentTransportID' and 'id' attribute is one-to-one.

The terminal SHALL comply with the following rules on maintaining the binding between fragment identifiers and transport identifiers in the terminal:

- If a terminal has received a Service Guide Delivery Descriptor (SGDD), the terminal SHALL record the binding between the ‘fragmentTransportID’ fields and the corresponding ‘fragmentID’ fields of the ‘Fragment’ element in that descriptor.
- If a terminal has not encountered the binding for a particular ‘fragmentTransportID’ in the SGDD, the terminal SHALL decode the corresponding fragment delivered in the SGDU, associate the decoded fragment identifier with the ‘fragmentTransportID’ in the SGDU header and record that binding.
- A terminal SHALL NOT use any recorded ‘fragmentID’-to-‘fragmentTransportID’ binding after the fragment expires.

#### 5.4.1.2 Resolving fragment references

Some Service Guide fragments have a capability to refer to other Service Guide fragments. This is achieved by including a reference element to the referring fragment. The element is named ‘<fragment name>Reference’ and the value of its attribute ‘idRef’ is of type ‘anyURI’ – this value contains the identifier of the referred fragment. The terminal can obtain the referenced fragment in two ways either receiving it over broadcast channel or retrieving it over interactive channel.

- When receiving the referred fragment over broadcast channel the ‘id’ attribute of the referred fragment is resolved to ‘transportID’ and consequently to ‘fragmentTransportID’ within the SGDU as specified in section 5.4.1.
- When retrieving the referred fragment over interactive channel, the delivery method as specified in section 5.4.3 (the case when a fragment is requested using the individual fragment ID) is used. In this context, the referred fragment id is used as a value in a name-value pair of attribute ‘fragmentID’ in the outgoing interactive request message.

All the terminals SHALL support receiving the referred fragment(s) over broadcast channel, while all terminals able to access the interaction channel SHALL support requesting and receiving the referred fragment(s) over the interaction channel also.

#### 5.4.1.3 Service Guide fragment encapsulation

In order to deliver the fragments from the network to the terminals the network needs to be able to place the fragments into the underlying transport frames. The network is provided with means of delivering more than one fragment as a atomic unit at one time but on the other hand the network is not restricted to deliver all the fragments at one go either. For the terminals to correctly receive and process any collection of fragments as one delivery unit the network SHALL comply with the following:

- The Service Guide Delivery Unit structure as defined in Table 1 SHALL be used for encapsulating Service Guide fragments for transport.
- The field ‘fragmentTransportID’ SHALL be assigned with the ‘transportID’ values as defined in section 5.4.1.1 to identify each of the fragments carried in the Service Guide Delivery Unit.
- When encapsulating the fragments into the Service Guide Delivery Unit, the mapping defined in section 5.4.1.1 SHALL be used.
- In case the SGDUs are listed in any FDT Instances the corresponding ‘Content-Type’ attributes SHALL be set to “application/vnd.oma.bcast.sgdu” to describe that the transport object contains an SGDU.

Using the ‘fragmentTransportID’ and ‘fragmentVersion’ fields the terminal can quickly infer whether the associated fragment in the SGDU has changed.

Data Field Name	Data Type
Service_Guide_Delivery_Unit {	
Unit_Header {	
extension_offset	uimsbf32
reserved	16 bits
n_o_service_guide_fragments	uimsbf24
for(i=0; i< n_o_service_guide_fragments; i++) {	
fragmentTransportID[i]	uimsbf32
fragmentVersion[i]	uimsbf32

offset[i]	uimsbf32
}	
}	
Unit_Payload {	
for(i=0; i<n_o_service_guide_fragments; i++) {	
fragmentEncoding[i]	uimsbf8
if(fragmentEncoding[i]=0) {	
fragmentType	uimsbf8
XMLFragment	bytestring
}	
else if(fragmentEncoding[i]=1) {	
validFrom	uimsbf32
validTo	uimsbf32
fragmentID	bytestring
SDPfragment	bytestring
}	
else if(fragmentEncoding[i]=2) {	
validFrom	uimsbf32
validTo	uimsbf32
fragmentID	bytestring
USBDFragment	bytestring
}	
else if(fragmentEncoding[i]=3) {	
validFrom	uimsbf32
validTo	uimsbf32
fragmentID	bytestring
ADPfragment	bytestring
}	
}	
}	
if(extension_offset>0) {	
extension_type	uimsbf8
next_extension_offset	uimsbf32
extension_data	bitstring
}	
}	

**Table 1: Service Guide Delivery Unit structure**

uimsbfN	Unsigned Nbit Integer, most significant bit first
bytestring	Array of bytes, each occupying eight bits
bitstring	Array of bits, length is multiple of eight

**Table 2: Mnemonics used in Table 1**

extension_offset	Offset in bytes from the start of the Unit_Payload to the start of the first extension. Set to 0 if there is no extension Present.
reserved	A bitfield reserved for future extensions of BCAST. All bits in this



	field SHALL be set to 0 in an SGDU conforming to this specification. Terminals MAY choose to ignore this field.
n_o_service_guide_fragments	Number of Service Guide fragments encapsulated in this specific Delivery Unit.
offset[i]	Offset in bytes from the start of the Unit_Payload to the start of the fragment_encoding field of the i:th Service Guide fragment. The offset list is sorted in ascending order.
fragmentTransportID[i]	Signals the identifier of the i:th Service Guide fragment which is defined for transport (see 5.4.1.5)
fragmentVersion[i]	Signals the version of the i:th Service Guide fragment. Note: The scope of the fragmentVersion is limited to this transport session. The value of fragmentVersion can turn over from 2 <sup>32</sup> -1 to 0.
fragmentEncoding[i]	Signals the encoding of the i:th Service Guide fragment, with the following values: 0 – XML encoded OMA BCAST Service Guide fragment 1 – SDP fragment 2 – MBMS User Service Bundle Description (USBD) as specified in [3GPP TS 26.346] (see 5.1.2.4 ‘SessionDescription’ element) 3 – XML encoded Associated Delivery Procedure as specified in [BCAST12-Distribution] section 5.3.4. 4-127 – reserved for future BCAST extensions 128-255 – available for proprietary extensions
fragmentType[i]	This field signals the type of an XML encoded BCAST Service Guide fragment, with the following values: 0 – unspecified 1 – ‘Service’ Fragment 2 – ‘Content’ fragment 3 – ‘Schedule’ Fragment 4 – ‘Access’ Fragment 5 – ‘PurchaseItem’ Fragment 6 – ‘PurchaseData’ Fragment 7 – ‘PurchaseChannel’ Fragment 8 – ‘PreviewData’ Fragment 9 – ‘InteractivityData’ Fragment 10-127 – reserved for BCAST extensions 128-255 – available for proprietary extensions
fragmentID	Null-terminated string containing the fragment ID of an SDP or MBMS USBD or Associated Delivery Procedure fragment as referenced from an ‘Access’ fragment via SessionDescriptionReference. Note: for an XML encoded OMA BCAST Service Guide fragment, this information is contained in the fragment itself.
validFrom	32 bit word representation of the validFrom value of an SDP or MBMS USBD or Associated Delivery Procedure fragment. This field is expressed as the first 32bits integer part of NTP time stamp. When set to “0” the interpretation is that “validFrom” is undefined. Note: for an XML encoded OMA BCAST Service Guide fragment, this information is contained in the fragment itself.
validTo	32 bit word representation of the validTo value of an SDP or MBMS USBD or Associated Delivery Procedure Description

	fragment. This field is expressed as the first 32bits integer part of NTP time stamp. When set to “0” the interpretation is that “validTo” is undefined. Note: for an XML encoded OMA BCAST Service Guide fragment, this information is contained in the fragment itself.
XMLfragment	String containing the actual XML data of the encapsulated Service Guide fragment without the termination character.
SDPfragment	String containing the actual SDP data, without termination character.
USBDfragment	String containing the actual MBMS USBD data, without termination character.
ADPfragment	String containing the actual XML data of the encapsulated Associated Delivery Procedure fragment, without termination character.
extension_type	Signals the type of the extension. 0-127 – reserved for BCAST extensions 128-255 – available for proprietary extensions Terminals MAY discard unknown extensions. In any case, they SHALL NOT get into an error state when they encounter unknown extensions.
next_extension_offset	Offset in bytes from the start of the current extension to the start of the next extension. Set to 0 if there is no next extension. The start of an extension is assumed to be the position of the extension type.
extension_data	Content of the extension.

Table 3: Semantics for Table 1

#### 5.4.1.4 Compression of Service Guide Delivery Units and Service Guide Delivery Descriptors

The network is provided with means of reducing the size of the SGDUs and SGDDs being delivered to terminals by using GZIP compression. For the algorithms and their respective signalling there are the following rules and constraints:

When FLUTE is used for transmission of the SDGUs

- the network MAY compress the SGDUs with the GZIP algorithm,
- terminals SHALL support both plain SGDUs and GZIP compressed SGDUs,
- When GZIP compression is used and the SGDUs are listed in any FDT Instances the corresponding ‘Content-Encoding’ attributes SHALL be set to “gzip”.

Additionally, when FLUTE is used for transmission of the SGDD, the network MAY compress the SGDDs with the GZIP algorithm. In this case the Content-Encoding attribute in the corresponding description of the FDT SHALL be set to “gzip”.

When ALC/LCT is used for transmission of the SGDUs

- the network MAY compress the SGDUs with the GZIP algorithm,
- terminals SHALL support both plain SGDUs and GZIP compressed SGDUs,
- When GZIP compression is used, the network SHALL signal GZIP compression of SGDUs by including the EXT\_CENC header in the ALC packet of the SGDU.

When HTTP is used for service guide delivery, the network MAY compress the HTTP response body with the GZIP algorithm. In this case the Content-Encoding attribute in the corresponding description of the HTTP response SHALL be set to “gzip”.

### 5.4.1.5 Service Guide Delivery Descriptor

Recall that all the fragments of the Service Guide are not necessarily placed into one and single SGDU but rather into multiple SGDUs. Even though the details of the exact delivery mechanisms are discussed in the later section, one needs to take one fundamental difference between the broadcast and interactive delivery into account here. This is the strategy used to divide the set of Service Guide fragments into transport layer subsets (each carried by its own SGDUs). In the broadcast case the division is fixed at a certain point of time and the division shared by the network and receiving terminals. In the interactive case there is no fixed division, the network simply provides one subset per request the subset contents being defined by the request.

In the broadcast case the division of fragments into separate transport frames (SGDUs) requires signalling of the details of the division, basically the SGDUs being used, how the SGDUs can be accessed on the transport and a map telling which fragment can be found in which SGDU. This transport layout signalling is provided by the network using the aforementioned SGDDs, Service Guide Delivery Descriptors, the signalling clearly representing transport dependent part of the information in the SGDDs.

Even though the details of the fixed division of the fragments into SGDUs are not applicable on the interactive delivery case, both the terminals performing the interactive or the broadcast retrieval of the Service Guide, however, rely on the SGDDs to provide exhaustive list of the fragments in the Service Guide.

In addition to dividing the Service Guide fragments into subsets for the transport layer, the network can also divide the fragments into subsets on the service layer independent on the transport layer division. This latter type of division of the Service Guide in the application layer is called grouping of the Service Guide and it is also signalled in the SGDDs. This concept, however, depends on application layer constructs and it is discussed in its own section.

When FLUTE is used for transmission of the SGDD, the network MAY compress the SGDDs with the GZIP algorithm. In this case the Content-Encoding attribute in the corresponding description of the FDT SHALL be set to "gzip". If HTTP is used for transmission of the SGDD and SGDU, the Content-Encoding attribute in the corresponding description of the HTTP response SHALL be set to "gzip".

Next section describes in more detail the transport dependent uses of the SGDD.

#### 5.4.1.5.1 Transport dependencies

Similarly as in the case of the network using more than one SGDU frame for delivering the Service Guide fragments, the network can also use multiple SGDDs for declaring the exhaustive list of the fragments in the Service Guide. In such a case it is easy to see that in order for the terminals to be aware of all the Service Guide fragments, the terminals need to be aware of all the SGDDs the network uses. For the broadcast delivery of the SGDDs, the network SHALL therefore place all the SGDDs representing a Service Guide into one and only one delivery session. This session is called the Service Guide Announcement Channel. The network SHALL also make sure that the SGDDs declare all fragments that are delivered over the broadcast channel.

As mentioned before for interactive delivery of the service guide, the main role of the SGDD is to declare all fragments that describe one or more services. The information about division of the fragments into SGDUs in this case is not essential, since all fragments are retrieved interactively and individually for each terminal, and thus a fixed division into SGDUs does not exist. However, the grouping in the service layer can be used to provide information about fragments belonging to the same service. The SGDD MAY declare fragments that are delivered over the interaction channel, but it SHALL at least declare a set of fragments that allow interactive retrieval of the complete SG. For example, the SGDD could declare only 'Service' fragments. The terminal could then interactively retrieve fragments related to specific selected services, using the request mechanism described in section 5.4.3.

#### 5.4.1.5.2 Service Guide Delivery Descriptor

The following structure SHALL be used for declaring availability, metadata and grouping of the fragments of Service Guide, and for delivery of the SG to point to the actual delivery channel and the delivery unit where the declared fragments are to be delivered. The terminal SHALL support the Service Guide Delivery Descriptor syntax as defined by XML Schema in [BCAST12-Schema-sg-sgdd]. For delivery, the Service Guide Delivery Descriptor SHALL be instantiated as an XML document.

The network is provided with means of reducing the size of the SGDDs being delivered to terminals by compressing SGDDs. For the algorithms and their respective signalling there are the following rules and constraints:

- the network MAY compress the SGDDs with the GZIP algorithm,
- terminals SHALL support both plain SGDDs and GZIP compressed SGDDs,

In the case of broadcast delivery of SGDDs, the network SHALL signal GZIP compression of SGDDs by setting the ‘content-encoding’ attribute of the FLUTE FDT instances listing the SGDDs. In the case of interactive delivery of SGDDs and SGDUs using HTTP, the network SHALL signal GZIP compression setting the ‘Content-Encoding’ attribute of the HTTP response.

Name	Type	Category	Cardinality	Description	Data Type
ServiceGuideDeliveryDescriptor	E			The Service Guide Delivery Descriptor Contains the following attributes: id version Contains the following elements: NotificationReception BSMLList DescriptorEntry TerminalCapability SGEntryPoints RMS AudienceMeasurement PrivateExt	
id	A	NM/TM	0..1	Unique identifier of the SGDD within one specific SG This attribute SHALL be instantiated if the SGDD is delivered over broadcast channel	anyURI
version	A	NM/TM	0..1	Version of SGDD. The newer version overrides the older one as soon as it has been received. This attribute SHALL be instantiated if the SGDD is delivered over broadcast channel	unsignedInt

Notification Reception	E1	NO/TO	0..1	<p>Reception information for general Notification Messages.</p> <p>In case of delivery over Broadcast channel, IPBroadcastDelivery specifies the address information for receiving Notification message.</p> <p>In case of delivery over Interaction channel, PollURL specify address information for polling notification and 'PollPeriod' specifies the associated polling period.</p> <p>When the Notification Message resource pointed by this element provides Notification Messages carrying Service Guide update, those SHALL relate to the currently bootstrapped Service Guide.</p> <p>If this element is present, at least one of the elements "IPBroadcastDelivery", or "PollURL" SHALL be present.</p> <p>This element SHALL be supported by the Network in case it supports the Notification function. Similarly, this element SHALL be supported by the Terminal in case it supports the Notification function.</p> <p>Contains the following elements: IPBroadcastDelivery PollURL PollPeriod</p>	
IPBroadcast Delivery	E2	NM/TM	0..1	<p>Provides IP multicast address and port number for reception of Notification Messages over the broadcast channel.</p> <p>Contains the following attributes: port address</p>	
port	A	NM/TM	1	General Notification Message delivery UDP destination port number; delivery over Broadcast Channel.	unsignedInt
address	A	NM/TM	1	General Notification Message delivery IP multicast address; delivery over Broadcast Channel.	string
PollURL	E2	NM/TM	0..N	<p>URL through which the terminal can poll general Notification Messages over Interaction Channel.</p> <p>If there are multiple instances of "PollURL" signaled, the terminal SHALL balance polling requests, within the set of "PollURL". Further, after having selected randomly, at a given time, a given URL, the terminal SHOULD use it for a while to benefit from cache management information as specified in HTTP 1.1 [RFC 2616] as it is reminded that this information is scoped to one given URL.</p>	anyURI

PollPeriod	E2	NO/ TM	0..1	<p>While polling the Notification Messages, the NTC is expected to poll every "PollPeriod" seconds.</p> <p>When this attribute is instantiated, no caching mechanisms of HTTP 1.1 [RFC 2616] SHOULD conflict with the fact that the NTC is expected to request for a fresh set of Notification Messages every "PollPeriod" value.</p> <p>The unit of this attribute is seconds</p>	decimal
BSMList	E1	NM/TM	0..1	<p>Declaration of the BSM Selectors which can be used in the GroupingCriteria sections defined below.</p> <p>Contains the following element: BSMSelector</p>	

BSMSelector	E2	NM/ TM	1..N	<p>Specifies the BSM associated with the fragments in this Service Guide Delivery Unit</p> <p>Allows a terminal to determine whether the SGDU's in this SGDD DescriptorEntry – among the SGDU's that are announced in various DescriptorEntries in various SGDD's – is associated with the terminal's affiliated BSM(s). The terminal's affiliated BSM(s) are represented within terminal as Management Objects with identifier '&lt;X&gt;/BSMSelector/BSMFilterCode' or as codes on the Smartcard as defined by [3GPP TS 22.022], [3GPP2 C.S0068-0], [3GPP TS 31.102], [3GPP2 C.], or [3GPP2 C. S0065-B]...</p> <p>For the interpretation of the BSMSelector within the SGDD the following SHALL apply:</p> <ul style="list-style-type: none"> <li>• If the BSMFilterCode present in this element matches to any of the '&lt;X&gt;BSMSelector//BSMFilterCode' entries within the terminal, or to any of the codes on the Smartcard, i.e. all of the instantiated attributes of BSMFilterCode have matching instantiated attributes under the '&lt;X&gt;/BSMFilterCode' or matching codes on the Smartcard, the terminal is able to process, render, interpret and handle the fragments without restrictions.</li> </ul> <p>Note that it is considered a match when the instantiated attributes under the BSMFilterCode matches a subset of the instantiated attributes of '&lt;X&gt;/BSMSelector/BSMFilterCode' or matches a subset of the codes on the SmartCard. However, when the instantiated BSMFilterCode represents a superset of attributes of the '&lt;X&gt;/BSMSelector/BSMFilterCode' or a superset of the codes on the Smartcard, it is not considered a match, because not all instantiated attributes under the BSMFilterCode matches to instantiated attributes of '&lt;X&gt;/BSMSelector/BSMFilterCode' or codes on the Smartcard. If the BSMFilterCode present in this element does not match to any of the '&lt;X&gt;/BSMSelector/BSMFilterCode' entries within the terminal, , i.e. not all of the instantiated attributes of BSMFilterCode have matching instantiated attributes under the '&lt;X&gt;/BSMSelector/BSMFilterCode' or codes on the Smartcard, the terminal can render, interpret and handle the fragments according to RoamingRules associated with this BSMSelector (identified by the attribute 'id'). In case the terminal does not have these RoamingRules the terminal SHALL NOT render the fragments to the user.</p> <p>The terminal MAY acquire the rules by sending a RoamingRuleRequest to</p>	
<p>© 2017 Open Mobile Alliance All Rights Reserved. Used with the permission of the Open Mobile Alliance under the terms as stated in this document.</p>				<p>[OMA-Template-Spec-20170101-I]</p>	

id	A	NM/TM	1	Identifier of the BSMSector. This 'id' is unique within network.	anyURI
roamingRuleRequestAddress	A	NO/TO	0..1	Address to which the terminals can send the RoamingRuleRequests to request RoamingRules associated with this BSMSector (identified with the 'id' attribute). Terminals supporting BroadcastRoaming SHALL support this attribute.	anyURI
BSMFilterCode	E3	NM/TM	0..1	The code that specifies this BSMSector. Contains the following attributes: type serviceProviderCode corporateCode serviceProviderName nonSmartCardCode Contains the following elements: NetworkCode3GPP NetworkCode3GPP2 Note: At most either 'NetworkCode3GPP' or 'NetworkCode3GPP2' SHALL be present. Implementation in XML Schema should use <choice>.	
type	A	NM/TM	1	The type of bsmFilterCode. 1 – BSMCode (Smart Card Code) This is used if the determination is made based on the country and operator code in the (U)SIM/(R-)UIM/CSIM 2 – BSMCode (Non Smart Card Code): This is used if the determination is made based on the country and operator code in the terminal Other values are reserved.	unsignedByte
serviceProviderCode	A	NO/TM	0..1	Service Provider Code as specified by [3GPP TS 22.022] or [3GPP2 C.S0068-0]. Applicable only when "type" == 1	unsignedByte
corporateCode	A	NO/TM	0..1	Corporate Code as specified by [3GPP TS 22.022] or [3GPP2 C.S0068-0]. Applicable only when "type" == 1	unsignedByte
serviceProviderName	A	NO/TM	0..1	Service Provider Name (SPN) as specified by [3GPP TS 31.102], [3GPP2 C.S0023-D], or [3GPP2 C. S0065-B]. Applicable only when "type" == 1	string
nonSmartCardCode	A	NO/TM	0..1	Value of BSMFilterCode when "type" == 2	string



NetworkCode3GPP	E4	NO/TM	0..1	IMSI-based Network, Network Subset or SIM/USIM codes as specified by [3GPP TS 22.022]. Applicable only when “type” == 1. Contains the following attributes: <ul style="list-style-type: none"> <li>- mobileCountryCode</li> <li>- mobileNetworkCode</li> <li>- networkSubsetCode</li> <li>- networkSubsetCodeRangeStart</li> <li>- networkSubsetCodeRangeEnd</li> <li>- codeRangeStart</li> <li>- codeRangeEnd</li> </ul>	
mobileCountryCode	A	NO/TM	0..1	Mobile Country Code (3 digits) as specified by [3GPP TS 22.022].	string of 3 digits
mobileNetworkCode	A	NO/TM	0..1	Mobile Network Code (2-3 digits) as specified by [3GPP TS 23.003].	string of 2-3 digits
networkSubsetCode	A	NO/TM	0..1	Network Subset Code (2 digits) as specified by [3GPP TS 22.022].	string of 2 digits
networkSubsetCodeRangeStart	A	NO/TM	0..1	Instead of providing an explicit code in attribute ‘networkSubsetCode’, the network MAY instead provide a continuous range of codes. In such a case the network SHALL <ul style="list-style-type: none"> <li>• provide the smallest code for the terminal to accept in this attribute,</li> <li>• the greatest code in the attribute ‘networkSubsetCodeRangeEnd’ and</li> <li>• SHALL not instantiate attribute ‘networkSubsetCode’.</li> </ul> The terminal SHALL interpret all the code values between the smallest and the greatest code as values to be accepted.	string of 2 digits
networkSubsetCodeRangeEnd	A	NO/TM	0..1	This attribute signals the end of the range of Network Subset Codes as specified above.	string of 2 digits
codeRangeStart	A	NO/TM	0..1	This attribute signals the lowest code value from a continuous range of one or more codes, which is used by the BCAST Terminal to determine whether a match exists with the local SIM/USIM code. The Terminal SHALL accept all code values between (and inclusive of) the lowest and highest code value for matching against the local SIM/USIM code.	string of 8 digits
codeRangeEnd	A	NO/TM	0..1	This attribute signals the highest code value for the BCAST Terminal to be considered valid for matching against the local SIM/USIM code, as described above.	string of 8 digits

NetworkCode3GPP2	E4	NO/TM	0..1	<p>IMSI and/or NAI based Network or (R-)UIM/CSIM codes as specified by [3GPP2 C.S0068-0]. Applicable only when “type” == 1. Contains the following attributes:</p> <ul style="list-style-type: none"> <li>- mobileCountryCode</li> <li>- mobileNetworkCode</li> <li>- iRMBasedMIN</li> <li>- hRPDRealm</li> <li>- ruimCSIMCodeRangeStart</li> <li>- ruimCSIMCodeRangeEnd</li> </ul>	
mobileCountryCode	A	NO/TM	0..1	Mobile Country Code (3 digits) as specified for Network Type 1 by [3GPP2 C.S0068-0].	string of 3 digits
mobileNetworkCode	A	NO/TM	0..1	Mobile Network Code (2-3 digits) as specified for Network Type 1 by [3GPP2 C.S0068-0].	string of 2-3 digits
iRMBasedMIN	A	NO/TM	0..1	First 4 digits of IRM-based MIN as specified for Network Type 2 by [3GPP2 C.S0068-0].	string of 4 digits
hRPDRealm	A	NO/TM	0..1	REALM code of the relevant HRPD network as specified by [3GPP2 C.S0068-0].	integer
ruimCSIMCodeRangeStart	A	NO/TM	0..1	<p>(R-)UIM or CSIM code, as specified in [3GPP2 C.S0023-D], [3GPP2 C. S0065-B] or [3GPP2 C.S0068-0].</p> <p>This attribute signals the lowest code value from a continuous range of one or more codes, which is used by the BCAST Terminal to determine whether a match exists with the local (R-)UIM/CSIM code. The Terminal SHALL accept all code values between (and inclusive of) the lowest and highest code value for matching against the local (R-)UIM/CSIM code.</p>	string
ruimCSIMCodeRangeEnd	A	NO/TM	0..1	<p>(R-)UIM or CSIM code, as specified in [3GPP2 C.S0023-D], [3GPP2 C. S0065-B] or [3GPP2 C.S0068-0].</p> <p>This attribute signals the lowest code value from a continuous range of one or more codes, which is used by the BCAST Terminal to determine whether a match exists with the local (R-)UIM/CSIM code. The Terminal SHALL accept all code values between (and inclusive of) the lowest and highest code value for matching against the local (R-)UIM/CSIM code.</p>	string
Name	E3	NM/TM	1..N	<p>Provides a user readable name for the BSM_Selector, possibly in multiple languages. The language is expressed using built-in XML attribute xml:lang with this element.</p> <p>This element can be used to provide information to the user so he can select the BSMSelector the terminal has to use.</p>	string

RoamingRule	E3	NO/TO	0..N	<p>Specifies a Roaming Rule associated with BSMSelector. The Roaming Rule specified by this element is not specific to any Home BSM. Such Roaming Rule can be interpreted as generic to any Home BSM, although the validity of the Roaming Rule for a particular pair of Visited BSM (as declared by the parent 'BSMSelector' element) and Home BSM is not guaranteed.</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>allowAll</li> <li>denyAll</li> </ul> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>TimeFrame</li> <li>AllowPurchaseItem</li> <li>AllowPurchaseData</li> <li>AllowService</li> <li>AllowContent</li> <li>DenyPurchaseItem</li> <li>DenyPurchaseData</li> <li>DenyService</li> <li>DenyContent</li> </ul> <p>Terminals supporting Broadcast Roaming SHALL support this element.</p> <p>The terminal SHALL interpret RoamingRule for each fragment so that in case 'allow' rule and 'deny' rule apply simultaneously, the 'deny' rule takes precedence.</p>	
allowAll	A	O	0..1	<p>Rule that, when set to "true", allows the Terminal to use all the fragments associated with BSMFilterCode associated with these RoamingRules.</p> <p>The default value of this attribute is "false".</p> <p>This attribute SHALL not be present if attribute 'denyAll' is present.</p>	boolean
denyAll	A	O	0..1	<p>Rule that, when set to "true", prohibits the Terminal to use any the fragments associated with BSMFilterCode associated with these RoamingRules.</p> <p>The default value of this attribute "false".</p> <p>This attribute SHALL not be present if attribute 'allowAll' is present.</p>	boolean
TimeFrame	E4	O	0..N	<p>Rule that defines the time frame(s) this RoamingRule is applies to.</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>startTime</li> <li>endTime</li> </ul>	
startTime	A	O	0..1	<p>Start of the time frame. If not given, the time frame is assumed to have started at some time in the past. This field is expressed as the first 32bits integer part of NTP time stamps.</p>	unsignedInt

endTime	A	O	0..1	End of the time frame. If not given, the time frame is assumed to end at some time in the future. This field is expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
Allow PurchaseItem	E4	O	0..1	Rule that allows the Terminal to use the listed PurchaseItems. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents GlobalPurchaseItemID that is allowed to be interpreted, rendered and accessed.	anyURI
Allow PurchaseData	E4	O	0..1	Rule that allows the Terminal to use the listed PurchaseData items. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents PurchaseData fragment ID that is allowed to be interpreted, rendered and accessed.	anyURI
Allow Service	E4	O	0..1	Rule that allows the Terminal to use the fragments corresponding to listed globalServiceIDs. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents globalServiceID. Fragments associated with this globalServiceID are allowed to be interpreted, rendered and accessed.	anyURI
Allow Content	E4	O	0..1	Rule that allows the Terminal to use the fragments corresponding to listed globalContentIDs. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents globalContentID. Fragments associated with this globalContentID are allowed to be interpreted, rendered and accessed.	anyURI
Deny PurchaseItem	E4	O	0..1	Rule that denies the Terminal to use the listed PurchaseItems. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	

Id	E5	M	1..N	This element contains value that represents globalPurchaseItemID that is denied to be interpreted, rendered and accessed..	anyURI
Deny PurchaseData	E4	O	0..1	Rule that denies the Terminal to use the listed PurchaseData items. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents PurchaseData fragment ID that is denied to be interpreted, rendered and accessed..	anyURI
Deny Service	E4	O	0..1	Rule that denies the Terminal to use the fragments corresponding to listed globalServiceIDs. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents globalServiceID. Fragments associated with this globalServiceID are denied to be interpreted, rendered and accessed.	anyURI
Deny Content	E4	O	0..1	Rule that denies the Terminal to use the fragments corresponding to listed globalContentIDs. This element SHALL not be present if one of “allowAll” or “denyAll” attribute is present. Contains the following element: Id	
Id	E5	M	1..N	This element contains value that represents globalContentID. Fragments associated with this globalContentID are denied to be interpreted, rendered and accessed.	anyURI
Notification Reception	E3	NO/TM	0..1	Reception information for Notification messages specific to the parent BSMSSelector. The terminal SHALL scope the information provided in Notification messages delivered via the children ‘IPBroadcastDelivery’ or ‘PollURL’ to the parent ‘BSMSSelector’. In case of delivery over Broadcast channel, IPBroadcastDelivery specifies the address information for receiving Notification message. In case of delivery over Interaction channel, PollURL specify address information for polling notification, and ‘PollPeriod’ specifies the associated polling period. If this element is present, at least one of the elements “IPBroadcastDelivery”, or “PollURL” SHALL be present. Contains the following elements: IPBroadcastDelivery PollURL PollPeriod	

IPBroadcast Delivery	E4	NM/TM	0..1	Provides IP multicast address and port number for reception of Notification Messages over the broadcast channel.  Contains the following attributes: port address	
port	A	NM/TM	1	BSM-specific Notification Message delivery UDP destination port number; delivery over Broadcast Channel.	unsignedInt
address	A	NM/TM	1	BSM-specific Notification Message delivery IP multicast address; delivery over the Broadcast Channel.	string
PollURL	E4	NM/TM	0..N	URL through which the terminal can poll Notification messages over the Interaction Channel.  If there are multiple instances of “PollURL” signaled, the terminal SHALL balance polling requests, within the set of “PollURL”. Further, after having selected randomly, at a given time, a given URL, the terminal SHOULD use it for a while to benefit from cache management information as specified in HTTP 1.1 [RFC 2616], as it is reminded that this information is scoped to one given URL.	anyURI
PollPeriod	E4	NO/TM	0..1	While polling the related service-specific Notification Messages, the NTC is expected to poll every “PollPeriod” seconds.  When this attribute is instantiated, no caching mechanisms of HTTP 1.1[RFC 2616] SHOULD conflict with the fact that the NTC is expected to request for a fresh set of Notification Messages every “PollPeriod” value.  The unit of this attribute is seconds	decimal
DescriptorEntry	E1	NM/TM	0..N	An entry in the Service Guide Delivery Descriptor.  Contains the following attribute: type  Contains the following elements: GroupingCriteria, Transport, AlternativeAccessURL, ServiceGuideDeliveryUnit  Note: A ‘simplified’ SGDD without any DescriptorEntry is only possible as a response result to a terminal request for a “simplified” SGDD (see section 5.4.3.3).	

type	A	NO/TM	0..1	<p>This attribute signals the partitioning strategy represented by this DescriptorEntry</p> <p>0: This DescriptorEntry represents a part of a set of fragments that is consistent (as defined in section 5.4.1.5.3). This means, further fragments of the same subset might be declared in other DescriptorEntries which share the same GroupingCriteria values with this DescriptorEntry. If this attribute is instantiated in one DescriptorEntry, it SHALL also be instantiated in all other DescriptorEntries contributing to the same set of fragments. It is reminded that more than one SGDD of the same Service Guide Announcement Channel can contribute to the same set of fragments as explained in section 5.4.1.5.3.</p> <p>1: This single DescriptorEntry represents a set of fragments that is consistent, i.e. all the fragments of the set are declared in this DescriptorEntry and do not reference other fragments outside this set</p> <p>2: This single DescriptorEntry represents a set of fragments that might not be consistent, i.e. some fragments in this set might reference other fragments declared in other DescriptorEntries which are using different GroupingCriteria values. Value 2 MAY be used in case one of the ‘TimeGroupingCriteria’, ‘BSMSelector’ or ‘TerminalCapabilityCriteria’ elements or all are instantiated, with the additional restrictions described in Section 5.4.1.5.3.</p> <p>3-255: Reserved for future use.</p> <p>The default value is 0.</p> <p>The network MAY support the attribute, but SHALL instantiate it if the DescriptorEntry is not of the default type 0.</p> <p>The terminal MAY use this attribute to determine if further DescriptorEntries contribute to a subset of fragments in addition to the current DescriptorEntry.</p>	unsignedByte
------	---	-------	------	--	--------------

GroupingCriteria	E2	NM/ TM	0..1	<p>Specifies the criteria for grouping Service Guide fragments in this Service Guide DescriptorEntry. If several criteria for grouping are present at the same time, all those grouping criteria apply to the set of Service Guide fragments in this Service Guide DescriptorEntry.</p> <p>Please note the same fragment may be declared in multiple DescriptorEntry of the same SGDD in case this fragment can meet multiple grouping criteria.</p> <p>Contains the following elements:</p> <p>TimeGroupingCriteria GenreGroupingCriteria BSMSelector ServiceCriteria TerminalCapabilityCriteria</p> <p>The 'GroupingCriteria' MAY be present as sub-element of 'DescriptorEntry' covering all fragments in this DescriptorEntry. Further, the 'GroupingCriteria' MAY be present as sub-element of 'Fragment' assigning the criteria to a particular fragment. When the 'GroupingCriteria' is present on both of the mentioned levels, the terminal SHALL consider the fragment-level 'GroupingCriteria' providing additional (not overriding) grouping criteria for the fragment.</p>	
TimeGroupingCriteria	E3	NM/ TM	0..1	<p>Specifies the period of time this DescriptorEntry describes. (For example: declares a certain subgroup of valid Service Guide fragments for next 2 hours). This field contains the 32bits integer part of an NTP time stamp.</p> <p>Contains the following attributes:</p> <p>startTime, endTime</p> <p>A fragment matches the TimeGroupingCriteria if it describes information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with the time interval specified by startTime/endTime.</p>	
startTime	A	NM/ TM	1	Start of the time period of TimeGroupingCriteria. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
endTime	A	NM/ TM	1	End of the time period of TimeGroupingCriteria. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt



GenreGroupingCriteria	E3	NM/ TM	0..1	<p>Specifies the classification of the services/content associated with the fragments in this Service Guide Delivery Unit (e.g. comedy, action, drama).</p> <p>The OMA BCAST Service Guide allows describing the format of the Genre element in the Service Guide in two ways:</p> <ul style="list-style-type: none"> <li>• The first way is to use a free string</li> <li>• The second way is to use the “href” attributes of the Genre element to convey the information in the form of a controlled vocabulary (classification scheme as defined in [TVA-Metadata] or classification list as defined in [MIGFG]).</li> </ul> <p>The built-in XML attribute xml:lang MAY be used with this element to express the language.</p> <p>The Network MAY instantiate several different sets of ‘Genre’ element, using it as a free string or with a ‘href’ attribute. The Network SHALL ensure the different sets have equivalent and non-conflicting meaning, and the terminal SHALL select one of the sets to interpret for the end-user.</p> <p>Contains the following attributes:</p> <p>type href</p>	string
type	A	NO/ TO	0..1	<p>This attribute signals the level of this ‘Genre’ element.</p> <p>The following values are allowed:</p> <p>“main” “secondary” “other”</p>	string

href	A	NO/ TO	0..1	<p>This attribute signals the controlled vocabulary used for this ‘Genre’ element.</p> <p>If this attribute is supported, the following applies to the support and use of classification schemes according to [TVA-Metadata]:</p> <ul style="list-style-type: none"> <li>for values of the ‘type’ attribute equal to "main" or "secondary", the terminal MAY support levels 1-4 of the TV Anytime ContentCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:ContentCS:2005 as defined in Annex A.8 of [TVA-Metadata]</li> <li>for a value of the ‘type’ attribute equal to "other", the terminal MAY support levels 1-3 of the TV Anytime IntendedAudienceCS classification scheme identified by the classificationSchemeURI urn:tva:metadata:cs:IntendedAudienceCS:2005 as defined in Annex A.11 of [TVA-Metadata]. When the IntendedAudienceCS is provided simultaneously with an instantiation of the ‘TargetUserProfile’, the two SHALL have equivalent meaning.</li> <li>The network SHALL use the following URI syntax to signal terms from classification schemes: &lt;classificationSchemeURI&gt; “:” &lt;termID&gt;</li> <li>If this attribute is instantiated by the network, the element ‘Genre’ SHALL be an empty string and the xml:lang attribute SHALL NOT be used.</li> </ul> <p>If this attribute is supported, the following applies to the support and use of the classification from [MIGFG]:</p> <ul style="list-style-type: none"> <li>This classification SHALL be signalled with the URI “http://www.loc.gov/rr/mopic/miggen.html”</li> <li>The string value carried in the ‘Genre’ element SHALL be used to convey the actual value of the classification as given in [MIGFG]</li> <li>The Network MAY use the values “main” and “secondary” of the ‘type’ attribute so as to provide an ordering of two classifications applying to the same Service.</li> </ul> <p>Other Classification Schemes MAY be signaled with the 'href' attribute, however how they are used is out of scope of this specification.</p> <p>If this attribute is not instantiated, the ‘Genre’ element SHALL be a free string.</p>	anyURI
------	---	-----------	------	--	--------

BSMSelector	E3	NM/ TM	0..N	Specifies the BSM associated with the fragments in this Service Guide Delivery Unit by referencing a BSMSelector structure declared above. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Reference to the identifier of the BSMSelector declared within the 'BSMList' above.	anyURI
ServiceCriteria	E3	NM/TM	0..1	Allows to group fragments by service. The value of this field is the fragment ID of the 'Service' fragment related to that service.	anyURI
TerminalCapabilityCriteria	E3	NO/ TM	0..N	Specifies the terminal capability requirements for the service consumption associated with the fragments in this Service Guide Delivery Unit by referencing a TerminalCapability structure declared above. Absence of this element implies the terminal can assume that it meets the capability requirements for service consumption associated with the fragments in this Service Guide Delivery Unit. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Reference to the identifier of the required terminal capabilities declared within the 'TerminalCapability' above.	anyURI
Transport	E2	NM/ TM	0..1	The pointer to the transport session delivering the Service Guide fragments within Service Guide Delivery Units announced in this DescriptorEntry. Contains the following attributes: ipAddress, port, srcIpAddress, transmissionSessionID, hasFDT	
ipAddress	A	NM/ TM	1	Destination IP address of the target delivery session	string
port	A	NM/ TM	1	Destination port of target delivery session	unsignedShort
srcIpAddress	A	NM/ TM	0..1	Source IP address of the delivery session In case source specific multicast scheme is applied in the transmission, then the 'srcIpAddress' attribute SHALL have as its value the IP address found in the IP-packets belonging to the IP-stream in question. In case this attribute is omitted, there SHALL only be one source IP address from which the file delivery session originates which is defined by the combination of destination IP address, port and transmission session ID given.	string
transmissionSessionID	A	NM/ TM	1	This is the Transmission Session Identifier (TSI) of the session at ALC/LCT level	unsignedShort

hasFDT	A	NO/ TM	0..1	<p>If FDT is transmitted in the transport session delivering the Service Guide fragments, this attribute SHALL be set to “true”. Otherwise this attribute SHALL be set to “false”. The default value of this attribute is “true”.</p> <p>If this element is set to “false”,</p> <ul style="list-style-type: none"> <li>the FEC parameters related to transport objects delivering SGDUs in the transport session SHALL be signalled using EXT_FTI [RFC 3926].</li> <li>the optional compression of SGDUs SHALL be signalled using EXT_CENC [RFC 3926]. Note that EXT_CENC was originally defined in [RFC 3926] for signaling the encoding of the FDT, but is assigned to a different usage in this specification for the specific case of SGDU delivery directly using ALC.</li> </ul>	boolean
AlternativeAccessURL	E2	NM/ TM	0..N	<p>Declares the alternative URL for retrieving the Service Guide fragments, declared in the parent ‘DescriptorEntry’ element, via the interaction channel. In addition, fragments not declared in the parent ‘DescriptorEntry’ MAY also be available. Terminal MAY check the availability of undeclared fragments by issuing an unspecific Service Guide request against the ‘AlternativeAccessURL’, as specified in section 5.4.3.2 of the present document.</p> <p>If there are multiple instances of AlternativeAccessURL signalled, the terminal SHALL randomly select one of them to use.</p> <p>Note: usage of this element is specified in section 5.4.1.5.4 of the present document.</p> <p>The terminal SHALL ignore ‘AlternativeAccessURL’ and SHALL use ‘UnicastServerURL’ instead, if there is a &lt;SGentrypoints&gt; element instantiated in the SGDD including at least one UnicastServerURL, and either scoped by no BSMselector or scoped by the BSMselector identifying the BSM which the terminal is affiliated to.</p>	anyURI
ServiceGuideDeliveryUnit	E2	NM/ TM	1..N	<p>A group of fragments.</p> <p>Contains the following attributes: transportObjectID, versionIDLength, contentLocation, validFrom, validTo</p> <p>Contains the following element: Fragment</p>	

transportObjectID	A	NM/TM	0..1	The transport object ID of the Service Guide Delivery Unit carrying the declared fragments within this group. If 'hasFDT' is assigned with value 'true', then the value of 'transportObjectID' SHALL match the value of the TOI paired in the FDT instance with the 'Content-Location' having as its value the value of the 'contentLocation' attribute below. If and only if element E2 'Transport' is instantiated, SHALL this attribute be instantiated.	positiveInteger
versionIDLength	A	NO/TO	0..1	Indicates the number of least significant bits representing the version ID in the transportObjectID, when Split TOI is used. If this element is omitted, the terminal assumes Split-TOI is not used.	unsignedLong
contentLocation	A	NM/TM	0..1	This is the location of the Service Guide Delivery Unit. It corresponds to the 'Content-Location' attribute in the FDT. If and only if element E2 'Transport' is instantiated, SHALL this attribute be instantiated.	anyURI
validFrom	A	NM/TM	0..1	The first moment of time this group of Service Guide fragments is valid. This field contains the 32bits integer part of an NTP time stamp. Note: If this attribute is not present, 'validFrom' attribute MUST be present in the 'Fragment' sub-element.	unsignedInt
validTo	A	NM/TM	0..1	The last moment of time this group of Service Guide fragments is valid. This field contains the 32bits integer part of an NTP time stamp. Note: If this attribute is not present, 'validTo' attribute MUST be present in the 'Fragment' sub-element.	unsignedInt
Fragment	E3	NM/TM	1..N	Declaration of Service Guide fragment. If the fragment is available over the broadcast channel it MUST be present here. If the fragment is available over the interaction channel it MAY be present here. Contains the following attributes: transportID, id version validFrom validTo fragmentEncoding fragmentType isAvailableOverIC Contains the following element: GroupingCriteria	
transportID	A	NM/TM	0..1	The identifier of the announced Service Guide fragment to be used in the Service Guide Delivery Unit header. Note: if the SG is delivered over the broadcast channel only, this element MUST be present	unsignedInt

id	A	NM/ TM	1	The identifier of the announced Service Guide fragment.	anyURI
version	A	NM/ TM	1	The version of the announced Service Guide fragment. Note: The scope of the version is limited to the given transport session. The value of version turn over from $2^{32}-1$ to 0.	unsignedInt
validFrom	A	NM/ TM	0..1	The first moment when this fragment is valid. If not given, the validity is assumed to have started at some time in the past. This field contains the 32bits integer part of an NTP time stamp. Note: If this attribute is present and 'validFrom' attribute of 'ServiceGuideDeliveryUnit' is also present, the value of this attribute overrides the value of 'ServiceGuideDeliveryUnit' attribute 'validFrom'.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when this fragment is valid. If not given, the validity is assumed to end in undefined time in the future. This field contains the 32bits integer part of an NTP time stamp. Note: If this attribute is present and 'validTo' attribute of 'ServiceGuideDeliveryUnit' is also present, the value of this attribute overrides the value of 'ServiceGuideDeliveryUnit' attribute 'validTo'.	unsignedInt
fragmentEncoding	A	NM/TM	1	Signals the encoding of a Service Guide fragment, with the following values: 0 – XML encoded OMA BCAST Service Guide fragment 1 – SDP fragment 2 – MBMS User Service Description as specified in [3GPP TS 26.346] (see 5.1.2.4, SessionDescriptionReference) 3 – XML encoded Associated Delivery Procedure as specified in [BCAST12-Distribution] section 5.3.4. 4-127 – reserved for future BCAST extensions 128-255 – available for proprietary extensions	unsignedByte

fragmentType	A	NM/TM	0..1	<p>This field signals the type of an XML encoded BCAST Service Guide fragment, with the following values:</p> <ul style="list-style-type: none"> <li>0 – unspecified</li> <li>1 – ‘Service’ Fragment</li> <li>2 – ‘Content’ fragment</li> <li>3 – ‘Schedule’ Fragment</li> <li>4 – ‘Access’ Fragment</li> <li>5 – ‘PurchaseItem’ Fragment</li> <li>6 – ‘PurchaseData’ Fragment</li> <li>7 – ‘PurchaseChannel’ Fragment</li> <li>8 – ‘PreviewData’ Fragment</li> <li>9 – ‘InteractivityData’ Fragment</li> <li>10-127 – reserved for BCAST extensions</li> <li>128-255 – available for proprietary extensions</li> </ul> <p>This attribute SHALL be present in case ‘fragmentEncoding’=0. Default: 0</p>	unsignedByte
isAvailableOverIC	A	NO/TM	0..1	<p>This attribute MAY be instantiated only in the case the fragment is broadcast and the broadcast SG is associated to a "complementary" entry point over the interaction channel, for the given BSM. This attribute declares whether this fragment is also available over the “complementary” entry point over IC. Based on this information, the terminal can decide to fetch this fragment over the interaction channel or over the broadcast channel depending on criteria such as terminal coverage, user/service provider preferences, etc.</p> <p>Note that this attribute can only be instantiated if this declaration applies to all the BSM(s) which the parent "DescriptorEntry" applies to.</p>	boolean
GroupingCriteria	E4	NM/TM	0..1	<p>Specifies the criteria for grouping this Service Guide fragment.</p> <p>If several criteria for grouping are present at the same time, all those grouping criteria apply to this fragment.</p> <p>Contains the following elements:</p> <ul style="list-style-type: none"> <li>TimeGroupingCriteria</li> <li>BSMSelector</li> </ul> <p>The ‘GroupingCriteria’ MAY be present as sub-element of ‘DescriptorEntry’ covering all fragments in this DescriptorEntry. Further, the ‘GroupingCriteria’ MAY be present as sub-element of ‘Fragment’ assigning the criteria to a particular fragment. When the ‘GroupingCriteria’ is present on both of the mentioned levels, the terminal SHALL consider the fragment-level ‘GroupingCriteria’ providing additional (not overriding) grouping criteria for the fragment.</p>	

TimeGroupingCriteria	E5	NM/TM	0..1	<p>Specifies the period of time as a grouping criteria for the fragment. (For example: declares a certain subgroup of valid Service Guide fragments for next 2 hours). This field contains the 32bits integer part of an NTP time stamp.</p> <p>Contains the following attributes:                      startTime                      endTime</p> <p>A fragment matches the TimeGroupingCriteria if it describes information related to content or interactivity that can be distributed, consumed or activated during a time interval that is not disjoint with the time interval specified by startTime/endTime.</p> <p>If a 'TimeGroupingCriteria' element exists at DescriptorEntry level which is applicable to this fragment, the time interval defined here at 'Fragment' level MUST NOT exceed the time interval defined at 'DescriptorEntry' level.</p>	
startTime	A	NM/TM	1	Start of the time period of TimeGroupingCriteria. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
endTime	A	NM/TM	1	End of the time period of TimeGroupingCriteria. This field contains the 32bits integer part of an NTP time stamp. Note: this is different than fragment validity time.	unsignedInt
BSMSelector	E5	NM/TM	0..N	<p>Specifies the BSM associated with this fragment by referencing a BSMSelector structure declared above.</p> <p>Contains the following attribute:                      idRef                      isAvailableOverIC</p>	
idRef	A	NM/TM	1	Reference to the BSMSelector declared within the BSMList structure above.	anyURI
isAvailableOverIC	A	NO/TM	0..1	This attribute MAY be instantiated only in the case when the fragment is broadcast and the broadcast SG is associated to a "complementary" entry point over the interaction channel, for the given BSM. This attribute declares whether this fragment is also available over the "complementary" entry point over IC. Based on this information, the terminal can decide to fetch this fragment over the interaction channel or over the broadcast channel depending on criteria such as terminal coverage, user/service provider preferences, etc. This attribute SHALL NOT be present, if it is already instantiated as an attribute of the parent "Fragment" E3 element.	boolean



TerminalCapability	E1	NO/TM	0..N	<p>Specifies the required terminal capabilities associated with the fragments in this Service Guide Delivery Unit</p> <p>This element provides the information to the terminal what is needed to consume the services declared by the SGDU's in this SGDD DescriptorEntry – among the SGDU's that are announced in various DescriptorEntries in various SGDD's.</p> <p>Based on this information the terminal can process the service guide fragments in dependence of the terminal capabilities. If the terminal can find a single match among the various distribution types under BroadcastServiceDeliveryType or UnicastServiceDeliveryType the terminal can decide to process the service guide fragments. In case the terminal cannot meet any of the specified capabilities requirements, the terminals can decide to acquire the service guide fragments declared by the SGDU's in this SGDD and just store them, or it can choose not to acquire them at all. In any case the terminal SHOULD NOT render the fragments associated with the service to the user, for which it cannot meet any of the specified terminal capability requirements. It has to be noted that terminal capabilities are not dependent on the local network coverage aspects, i.e. a terminal MAY decide to render the fragments associated to a given broadcast (resp. unicast) delivery system for which it is temporarily out of radio coverage.</p> <p>Contains the following elements: BroadcastServiceDeliveryType UnicastServiceDeliveryType</p>	
id	A	NM/TM	1	Identifier of the TerminalCapability. This 'id' is unique within network.	anyURI
BroadcastServiceDeliveryType	E2	NM/TM	0..N	<p>Identifier of the type of underlying broadcast distribution system that is required to consume the content and services. When multiple BroadcastServiceDeliveryType elements are instantiated, the terminal only requires support for one of the types in order to be able to consume the service.</p> <p>Contains the following element: Type Version</p>	

Type	E3	NM/ TM	0..1	Type of underlying broadcast distribution system, possible values: 0. IPDC over DVB-H 1. 3GPP MBMS 2. 3GPP2 BCMCS 3. DVB-SH 4. WiMax 5. FLO 6. DVB-NGH 7. DVB-T2 8 -127. reserved for future use 128 -255. reserved for proprietary use	unsignedByte
Version	E3	NM/ TM	0..N	Version of underlying broadcast distribution system. Possible values for MBMS are specified according to the following syntax in ABNF [RFC4234]:  version = "3GPP." release "." bearer  release = "R6" / "R7" / "R8"  bearer = "GERAN" / "UTRAN" / "MBSFN-FDD" / "MBSFN-TDD" / "MBSFN-IMB"  By applying these rules, strings such as 3GPP.R6.UTRAN and 3GPP.R8.MBSFN-IMB can be constructed. To allow backwards compatibility with BCAST 1.0, the strings "Rel-6" and "Rel-7" (which do not follow the ABNF syntax above) are also included in the set of possible values for MBMS.  Other possible values include, e.g., 1x or HRPD or Enhanced HRPD for BCMCS.	string
UnicastDeliveryType	E2	NM/TM	0..N	Identifier of the type of underlying unicast distribution system that is required to consume the content and services. When multiple UnicastServiceDeliveryType elements are instantiated, the terminal only requires support for one of the types in order to be able to consume the service.  Contains the following element: Type Version	
Type	E3	NM/ TM	0..1	Type of underlying unicast delivery system, possible values: 0. 3GPP unicast 1 -127. reserved for future use 128 -255. reserved for proprietary use	unsignedByte
Version	E3	NM/ TM	0..N	Version of underlying unicast delivery system. For instance, possible values are Rel-6 or Rel-7 for 3GPP.	string

SGEntryPoints	E1	NO/TM	0..N	<p>Each SGEntryPoints element declares entry points which are associated with one BSM selector or independent from any BSM selector. The SGDDs that are signalled in a given FLUTE FDT instance from a SG Announcement Channel SHALL contain the same set of SGEntryPoints elements.</p> <p>In the same SGDD, at most once SGEntryPoints element including a given BSMselector SHALL be instantiated</p> <p>In the same SGDD, at most once SGEntryPoints element including no BSMselector SHALL be instantiated</p> <p>This element SHALL not be instantiated in a response of a terminal request over the interaction channel, if the key "SGEntryPointsOnly" (see section 5.4.3.3) is not present in the request.</p> <p>Contains the following attribute: id</p> <p>Contains the following elements: BSMSelector SGEntryPoint</p> <p>Note that the support of the SGEntryPoints element is not mandated for non connected terminals</p>	
id	A	NM/TM	0..1	Identifies this particular "SGEntryPoints" element. This attribute is scoped by the announcement channel (resp. interactive bootstrap URL) from which the "SGEntryPoints" element is fetched. A given SGEntryPoints SHALL have a different id when it has been updated and SHALL not reuse the same id, except for "id" wrap-around case.	unsignedInt
BSMSelector	E2	NM/TM	0..1	<p>Specifies the BSM associated with the entry point by referencing a BSMSelector structure declared above.</p> <p>Note that if this element is not instantiated, then any terminal that fetches this given SGDD SHALL consider that the related SGEntryPoint applies to its affiliated BSM.</p> <p>Contains the following attribute: idRef</p>	
idRef	A	NM/TM	1	Reference to the identifier of the BSMSelector declared within the 'BSMList' above.	anyURI

SGEntryPoint	E2	NM/TM	1..N	<p>This element gives the SG entry points that relate to a given BSM over interaction and/or broadcast channels.</p> <p>Contains the following elements:</p> <p>BroadcastServerSession</p> <p>UnicastServerURL</p> <p>For each declared SGEntryPoint, BroadcastServerSession and/or UnicastServerURL(s) can be instantiated. When both are instantiated in a given SGEntryPoint, the list of UnicastServerURLs is the set of SG entry points over the interaction channel that are associated to the declared Broadcast Entry Point, bearing different possible relationships (see the attribute relationOfICWithBC of UnicastServerURL for more details). The terminal can select among the available and applicable entry points. The selection process (e.g., can depend on factors such as user or service provider preference) is out-of-scope of the current specification.</p>	
BroadcastServerSession	E3	NM/TM	0..1	<p>This element gives the SG entry point over broadcast channel.</p> <p>“BroadcastServerSession” SHALL not be instantiated when it is intended to refer to the actual entry point (i.e. Announcement Channel) over which the current SGDD has been delivered. When the declared unicast entry point(s) are independent from any broadcast entry point, i.e., the "relationOfICWithBC" attribute of UnicastServerURL is set to value “0”, “BroadcastServerSession” SHALL not be instantiated either.</p> <p>Contains the following attributes:</p> <p>ipAddress</p> <p>port</p> <p>srcIpAddress</p> <p>transmissionSessionID</p> <p>contains the following element:</p> <p>BDSType</p>	
ipAddress	A	NM/TM	1	Destination IP address of the target delivery session	string
port	A	NM/TM	1	Destination port of target delivery session	unsignedShort

srcIpAddress	A	NM/ TM	0..1	Source IP address of the delivery session In case source specific multicast scheme is applied in the transmission, then the 'srcIpAddress' attribute SHALL have as its value the IP address found in the IP-packets belonging to the IP-stream in question. In case this attribute is omitted, there SHALL only be one source IP address from which the file delivery session originates which is defined by the combination of destination IP address, port and transmission session ID given.	string
transmissionSessionID	A	NM/ TM	1	This is the Transmission Session Identifier (TSI) of the FLUTE session	unsignedShort
BDSType	E4	NM/ TM	1	Information about the underlying broadcast distribution system for SG delivery Contains the following attributes: type version Contains the following element: BDSSpecificEntryPointInfo	complexType
type	A	NM/TM	1	Type of underlying broadcast distribution system for SG delivery, possible values: 0. IPDC over DVB-H 1. 3GPP MBMS 2. 3GPP2 BCMCS 3. DVB-SH 4. WiMax 5. FLO 6. DVB-NGH 7. DVB-T2 8 -127. reserved for future use 128 -255. reserved for proprietary use	unsignedByte

version	A	NM/ TO	0..1	<p>Version of underlying broadcast distribution system. Possible values for MBMS are specified according to the following syntax in ABNF [RFC4234]:</p> <pre> version = "3GPP." release "." bearer release = "R6" / "R7" / "R8"  bearer = "GERAN" / "UTRAN" / "MBSFN-FDD" / "MBSFN-TDD" / "MBSFN-IMB"                     </pre> <p>By applying these rules, strings such as 3GPP.R6.UTRAN and 3GPP.R8.MBSFN-IMB can be constructed. To allow backwards compatibility with BCAST 1.0, the strings "Rel-6" and "Rel-7" (which do not follow the ABNF syntax above) are also included in the set of possible values for MBMS.</p> <p>Other possible values include, e.g., 1x or HRPD or Enhanced HRPD for BCMCS. When this attribute is omitted, it means that it applies to all versions.</p> <p>A terminal supporting the BDS signaled by "type" attribute SHALL support this attribute when present, and SHALL ignore it otherwise.</p>	string
BDSSpecificEntryPointInfo	E5	NM/TO	0..1	<p>The placeholder for the supplementary information that is required in order to retrieve the broadcast SG entry point in a given BDS (for example, platform_id for DVB-H). The use of this placeholder is described in BDS adaptation specifications</p> <p>A terminal supporting the BDS signaled by "type" attribute SHALL support this element (whatever the substituted type signalled by 'xsi:type' is), and SHALL ignore it otherwise.</p>	Abstract complexType "BDSSpecificEntryPointInfoType", derived by extension in respective BDS Adaptation TSs.
UnicastServerURL	E3	NM/TM	0..N	<p>This element gives the information regarding the SG entry point over the interaction channel. If there are multiple declared URLs, terminals will choose the appropriate one according to the purpose of accessing SG over the interaction channel.</p> <p>Contains the following attributes: relationOfICwithBC url contains the following element: UnicastType</p>	

relationOfIC WithBC	A	NM/TM	1	<p>The relation between the broadcast and unicast SG entry points. Depending on this information, the terminal can choose the appropriate SGEnterPoint to retrieve a SG exclusively over BC or exclusively over IC or a complete SG over both BC and IC.</p> <p>0 – the unicast SG is independent from any broadcast SG. In this case, the unicast SG content is not necessary the same as any broadcast SG and is typically independent of the BDS (when the attribute has this value, the element BroadcastServerSession is not instantiated).</p> <p>1 – the unicast SG complements the broadcast SG. In this case, the broadcast SG is a self-consistent SG while the SG over unicast adds some further fragments (e.g, interactivityData) onto the broadcast SG. Some SG fragments can be common between the broadcasted SG and the unicast SG. The terminal can choose to retrieve a SG exclusively over BC or a complete SG over both BC and IC, depending on criteria like user/service provider preferences, or terminal coverage.</p> <p>2—the unicast SG entry point is used to repair the SG transmitted over the broadcast entry point. This is typically used to provide a ‘backup’ entry point when the terminal loses broadcast network access.</p> <p>3 – The unicast SG is a superset of the broadcast SG. Both the broadcast SG and the unicast SG are self-consistent. Depending on criteria like user or services provider preferences, terminal coverage, the terminal can choose to obtain the SG exclusively over BC , exclusively over IC, partially over BC and partially over IC.</p> <p>4- 127: Reserved for future use 128 -255: Reserved for proprietary use</p> <p>For “repair” UnicastServerURL, the terminal can send request for specific fragments or SGDD identified by their ids (see clause 5.4.3.3 and 5.4.3.4); for “complementary”, “independent” and “superset” entry points over the interaction channel, the terminal can send request for (un)specific SGDDs and fragments as described in clause 5.4.3.2, 5.4.3.3 and 5.4.3.4.</p>	unsignedByte
url	A	NM/TM	1	The URL of the entry point over IC.	anyURI
UnicastType	E4	NM/TM	1	<p>Type of underlying unicast delivery system for SG, possible values:</p> <p>0. 3GPP unicast 1 -127. reserved for future use 128 -255. reserved for proprietary use</p>	unsignedByte
version	A	NM/TM	0..1	<p>Version of underlying unicast delivery system. For instance, possible values are Rel-6 or Rel-7 for 3GPP.</p>	string

RMS	E1	NO/TO	0..N	Signals the existence of Rich Media Solution template documents for the presentation of SG. If the terminal has delays in rendering the Rich Media Solution template, it may render the SG using its native rendering engine during the meantime. Contains the following elements: BSMSelector RMSTemplate	
BSMSelector	E2	NM/TM	0..1	Specifies the BSM associated with the RMSTemplate by referencing a BSMSelector structure declared above.  Note that if this element is not instantiated, then any terminal that fetches this given SGDD SHALL consider that the related RMSTemplate applies to its affiliated BSM. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Reference to the identifier of the BSMSelector declared within the 'BSMList' above.	anyURI
RMSTemplate	E2	NO/TM	1..N	Access details for retrieving Rich Media Solution template document. Contains the following attribute: templateVersion Contains the following elements: Criteria Capabilities Transport AlternativeURL	
templateVersion	A	NO/TM	0..1	The version of the template. If the template version is newer than the one stored on the terminal then the terminal is applicable to receive the RMS template.	unsignedInt
Criteria	E3	NO/TO	0..N	Declares the criteria required for receiving this template. This element is used to control the reception of the template in terminals. Detailed filtering mechanisms can be deployed by using the attributeName/Value combination. Contains the following attributes: attributeName attributeValue	
attributeName	A	NO/NM	1	Criteria attribute name	string
attributeValue	A	NO/TM	1	Criteria attribute value	string



Capabilities	E3	NO/ TM	1	Describes the type and complexity of Rich Media Solution the rich media engine has to deal with.	complexType as defined in section 5.1.2.4 for Capabilities element child of RichMedia element in Access fragment
Transport	E3	NM/ TM	0..1	The pointer to the transport session delivering the RMS template.  Contains the following attributes: ipAddress port srcIpAddress transmissionSessionID hasFDT transmissionObjectID contentLocation	
ipAddress	A	NM/TM	1	Destination IP address of the target delivery session	string
port	A	NM/TM	1	Destination port of target delivery session	unsignedShort
srcIpAddress	A	NM/ TM	0..1	Source IP address of the delivery session In case source specific multicast scheme is applied in the transmission, then the 'srcIpAddress' attribute SHALL have as its value the IP address found in the IP-packets belonging to the IP-stream in question. In case this attribute is omitted, there SHALL only be one source IP address from which the file delivery session originates which is defined by the combination of destination IP address, port and transmission session ID given.	string
transmissionSessionID	A	NM/ TM	1	This is the Transmission Session Identifier (TSI) of the session at ALC/LCT level	unsignedShort
hasFDT	A	NO/ TM	0..1	If FDT is transmitted in the transport session delivering the RMS template, this attribute SHALL be set to "true". Otherwise this attribute SHALL be set to "false". The default value of this attribute is "true". If this element is set to "false", (1) the FEC parameters related to transport objects delivering SGDUs in the transport session SHALL be signalled using EXT_FTI[RFC 3926] (2) the optional compression of SGDUs SHALL be signalled using EXT_CENC [RFC 3926]. Note that EXT_CENC was originally defined in [RFC 3926] for signaling the encoding of the FDT, but is assigned to a different usage in this specification for the specific case of SGDU delivery directly using ALC.	boolean

transmissionObjectID	A	NM/TM	1	The Transport Object ID of the RMS template. If 'hasFDT' is assigned with value 'true', then the value of 'transportObjectID' SHALL match the value of the TOI paired in the FDT instance with the 'Content-Location' having as its value the value of the 'contentLocation' attribute below.	positiveInteger
contentLocation	A	NM/TM	0..1	The location of the RMS template. It corresponds to the 'Content-Location' attribute in the FDT. If and only if attribute 'hasFDT' is instantiated, SHALL this attribute be instantiated.	anyURI
AlternativeURL	E3	NO/TM	0..1	Declares the alternative URL for retrieving the RMS template, declared in the parent 'RMSTemplate' element, via the interaction channel.	anyURI
AudienceMeasurement	E1	NO/TO	0..1	Signals the Audience Measurement function Contains the following elements: CampaignInfo AudienceMeasurementTrigger	
CampaignInfo	E2	NO/TO	0..N	Audience Measurement Campaign specific information. The Terminal SHALL support this element if it supports Audience-Measurement Trigger over unicast. This element SHALL be instantiated when Audience Measurement Trigger message over SMS bearer is used. Contains the following elements: BSMSelector CampaignID ServerAddressURL	
BSMSelector	E3	NO/TM	0..1	Specifies the BSM associated with the CampaignInfo by referencing a BSMSelector structure declared above. Note that if this element is not instantiated, then any Terminal that fetches this given SGDD SHALL consider that the related CampaignInfo applies to its affiliated BSM. Contains the following attribute: idRef	
idRef	A	NM/TM	1	Reference to the identifier of the BSMSelector declared within the 'BSMList' above.	anyURI
CampaignID	E3	NM/TM	1..N	Identifier for the Audience Measurement Campaign(s) this info belongs to. The CampaignID is unique within the SG, and considered valid as long as present in SGDD. A particular CampaignID SHALL NOT be present under more than one CampaignInfo element.	unsignedInt

ServerAddressURL	E3	NM/TM	1..N	Address of the Audience Measurement server to which the Audience Measurement Request message is sent. This information is provided to verify Audience Measurement server addresses given in the Audience Measurement Trigger over SMS bearer. This server URL can be used together with all of the campaigns whose CampaignID's are declared under same CampaignInfo element as this particular ServerAddressURL	anyURI
AudienceMeasurementTrigger	E2	NO/TO	0..N	Audience Measurement Trigger. This element that is instantiated when Audience Measurement campaign is triggered from Service Guide. The delivery of the Service Guide can be over broadcast or unicast channel. Contains the following attributes: campaignID randomSelector campaignStartTime campaignEndTime Contains the following elements: BSMSelector UserConsentInformation ServerAddressURL AdditionalInfoAddressURL	
campaignID	A	NM/TM	1	Identifier for the Audience Measurement Campaign.	unsignedInt
randomSelector	A	NM/TM	0..1	Value in percent describing the proportion of Terminals expected to participate the campaign out of all the Terminals that receive the AudienceMeasurementTrigger. A Terminal which receives this AudienceMeasurement Trigger SHALL generate a random number from 0 (exclusive) to 100 (inclusive) and compare it with the value of randomSelector; if the random number is less than or equal to the value of randomSelector, then the Terminal SHALL enter Opt-In process; or else, if the random number is bigger than the value of randomSelector, then the Terminal SHALL ignore this Invitation. Comparison is performed once for each AudienceMeasurement Trigger message. This attribute SHALL be instantiated when the Invitation is delivered over broadcast channel. Allowed values are from 0 (exclusive) to 100 (inclusive).	decimal

campaignStartTime	A	NM/TM	1	<p>Time when the measurement is planned to start.</p> <p>This parameter is directed to the Terminal for pre-filtering purposes, e.g. to filter overlapping campaigns if the Terminal is not capable of handling multiple campaigns simultaneously.</p> <p>This information SHALL NOT be used to configure the Audience Measurement function on the Terminal.</p> <p>This information SHOULD NOT be presented to the user.</p> <p>This field contains the 32-bits integer part of an NTP time stamp.</p>	unsignedInt
campaignEndTime	A	NM/TM	1	<p>Time when the measurement is planned to end.</p> <p>This parameter is directed to the Terminal for pre-filtering purposes, e.g. to filter overlapping campaigns if the Terminal is not capable of handling multiple campaigns simultaneously, or for expiring this information from the Terminal after campaignEndTime. This information SHALL NOT be used to configure the Audience Measurement function on the Terminal.</p> <p>This information SHOULD NOT be presented to the user.</p> <p>This field contains the 32-bits integer part of an NTP time stamp.</p>	unsignedInt
BSMSelector	E3	NO/TM	0..1	<p>Specifies the BSM associated with the AudienceMeasurementTrigger by referencing a BSMSelector structure declared above.</p> <p>Note that if this element is not instantiated, then any Terminal that fetches this given SGDD SHALL consider that the related AudienceMeasurementTrigger applies to its affiliated BSM.</p> <p>Contains the following attribute: idRef</p>	
idRef	A	NM/TM	1	Reference to the identifier of the BSMSelector declared within the 'BSMList' above.	anyURI
UserConsentInformation	E3	NM/TM	1	<p>User consent information needed to perform opt-in for campaign participation.</p> <p>Contains the following attributes: consentRequired</p> <p>Contains the following elements: CampaignName CampaignDescription</p>	

consentRequired	A	NM/TM	1	<p>User consent required for Campaign participation. If this attribute is FALSE the Terminal MAY perform silent Opt-In (“Refuse” or “Accept”).</p> <p>If this attribute is TRUE, the Terminal MAY perform a silent Opt-In, which MAY result “Refuse” (opted-out), but SHALL NOT result “Accept” (opted-in). If the Terminal did not silently refuse the invitation (“Refuse”), the Terminal SHALL consult the user for the participation in the campaign, and the result of the Opt-In process is according to the user choice (“Refuse” or “Accept”).</p>	boolean
CampaignName	E4	NM/TM	1..N	<p>Description of the campaign.</p> <p>This information SHALL be shown to the user, when user consent is to be requested.</p> <p>The language is expressed using built-in XML attribute ‘xml:lang’ with this element.</p>	string
CampaignDescription	E4	NM/TM	1..N	<p>Description of the campaign.</p> <p>This information is intended to be displayed to the user.</p> <p>In case the campaign requires user consent (i.e. ‘consentRequired’ is set to TRUE) and the Terminal did not silently refuse the invitation (“Refuse”), this information SHALL be shown to the user, and MAY include the Terms of Use of the campaign.</p> <p>The language is expressed using built-in XML attribute ‘xml:lang’ with this element.</p>	string
ServerAddressURL	E3	NM/TM	1..N	<p>Address of the Audience Measurement server to which the Audience Measurement Request message, as specified in [BCAST12-Services], is sent.</p> <p>If there are multiple instances of ServerAddressURL signalled, the Terminal SHALL randomly select one of them to use.</p> <p>Contains the following attributes: randomTime</p>	anyURI
randomTime	A	NM/TM	1	<p>The ‘randomTime’ refers to the time window length over which a Terminal SHALL calculate a random time for the transmission of interaction. The method provides for statistically uniform distribution over a relevant period of time.</p> <p>The Terminal SHALL calculate a uniformly distributed random time out of the interval between 0 and randomTime. The unit is seconds (fractions can be expressed using data type Decimal). ‘randomTime’ shall be a non-negative number.</p>	decimal

AdditionalInfoAddressURL	E3	NO/TO	0..N	URL for additional information related to this campaign If there are multiple instances of AdditionalInfoAddressURL signalled, the Terminal SHALL randomly select one of them to use.	anyURI
PrivateExt	E1	NO/TO	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	NO/TO	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.4.1.5.3 Grouping of Service Guide

One can divide the set of fragments comprising a Service Guide into subsets. These subsets, also referred to as groups, are formed by first selecting a criterion (or possibly criteria). Examples of a criterion are intervals of time, BSM codes and identifiers of ‘Service’ fragments. Once a criterion is selected, each subset (or group) to be formed is assigned a value from the domain of the criterion. The contents of each subset are then determined by comparing the value of the criterion represented by the subset to each of the fragments of the Service Guide. If and only if the values of the subset and the fragment match, the fragment is placed into the subset

For example, one could have a subset representing all the fragments representing the contents of the SG today from 10 a.m. to 12 a.m. In such a case the grouping criteria is an interval of time, the value represented by the subset is the interval from 10 o’clock to 12 o’clock and any fragment representing the contents of the Service Guide within that time frame belongs to the subset.

The network MAY

- use multiple grouping criteria simultaneously,
- form overlapping (i.e. mutually non-exclusive) subsets (or groups) and
- place the declaration(s) of the subsets or even a single subset into multiple SGDDs.

In the case of broadcasted SGDD(s), it is RECOMMENDED that the network creates self-contained sets of fragments, i.e. attempts not to place a fragment into a subset if the fragment contains a reference to a fragment not present in the same subset.

Values ‘0’ and ‘1’ of the ‘type’ attribute of the ‘DescriptorEntry’ element in ‘ServiceGuideDeliveryDescriptor’ allow declaration of self-contained sets of fragments. This restriction introduces the constraint of consistency on the Service Guide subsets (or groups). A Service Guide subset is consistent when it is self-contained, i.e. when no fragment from that subset references any fragment that is not in the said subset.

In case non-self-contained subsets are used, the network SHALL appropriately flag them using value ‘2’ for the ‘type’ attribute of ‘DescriptorEntry’ in ‘ServiceGuideDeliveryDescriptor’. In such case, it is however reminded that the network is still bound to describe the complete Service Guide in an exhaustive manner (see section 5.5.1.2) in the case the Service Guide is delivered over the broadcast channel. Consequently, the network SHALL ensure that the terminal can still build a self-contained set of fragments out of an actual ‘DescriptorEntry’  $X$  with the type attribute set to ‘2’ by looking for missing fragments in a defined subset  $S$  of all other ‘DescriptorEntry’ instances.

- 1) for time-based partitioning, the set  $S$  contains all those DescriptorEntries  $Y_i$  which signal in the ‘TimeGroupingCriteria’ element a start time value that is less than or equal to the start time value signalled in the ‘TimeGroupingCriteria’ of DescriptorEntry  $X$ .
- 2) for BSMSelector-based partitioning, the set  $S$  contains all those DescriptorEntries  $Y_i$  which signal in the ‘BSMSelector’ element a set of values that is a superset of the set of values signalled in the ‘BSMSelector’ element of DescriptorEntry  $X$ .

- 3) for terminal capability-based partitioning, the set  $S$  contains all those DescriptorEntries  $Y_i$  which signal in the 'TerminalCapability' element a broadcast/unicast distribution system that is
  - equal to the set of values signalled in the 'TerminalCapability' element of DescriptorEntry  $X$ .
  - or that refers to the broadcast/unicast distribution system over which the current SGDD is retrieved.
- 4) for a partitioning that uses a combination of the above three partitioning criteria, the set  $S$  is defined as the intersection of the sets corresponding to the criteria (1), (2) and (3).
- 5) for a partitioning that combines 'TimeGroupingCriteria', 'BSMSelector', 'TerminalCapabilityCriteria' or all with 'ServiceCriteria' and/or 'GenreGroupingCriteria', the set  $S$  is independently computed for each combination of 'ServiceCriteria' and 'GenreGroupingCriteria' values according to (1), (2), (3) or (4) above. This means that for a given DescriptorEntry  $X$ , the set  $S$  contains all those DescriptorEntries  $Y_i$  which satisfy the above criteria and signal the same values as  $X$  in the grouping criteria 'GenreGroupingCriteria' and/or 'ServiceCriteria'.

Note: this ensures that the terminal does not need to scan the full set of all DescriptorEntries to build said consistent subset from a DescriptorEntry of type '2', but that it can discover all needed fragments by traversing the DescriptorEntries in the set  $S$  one by one exploiting the (partial) order relationship defined by the above criteria. For example, in case there are 'ServiceCriteria' ("TV1", "TV2"), 'TimeGroupingCriteria' ("today", "tomorrow") and 'GenreGroupingCriteria' ("News", "Sports") instantiated, the terminal only needs to look into the DescriptorEntry  $Y_1$  with 'ServiceCriteria'="TV2", 'GenreGroupingCriteria'="News" and 'TimeGroupingCriteria'="today" in order to find the missing fragments for DescriptorEntry  $X$  with 'ServiceCriteria'="TV2", 'GenreGroupingCriteria'="News" and 'TimeGroupingCriteria'="tomorrow".

Furthermore, for example, in case there are 'ServiceCriteria' ("TV1", "TV2"), 'TimeGroupingCriteria' ("today", "tomorrow"), 'TerminalCapabilityCriteria' ("DVB-H", "DVB-SH", "FLO", "MBMS") and 'GenreGroupingCriteria' ("News", "Sports") instantiated for an ESG that is retrieved via a DVB-H network, the terminal only needs to look into the:

- and the DescriptorEntry  $Y_2$  with 'ServiceCriteria'="TV2", 'GenreGroupingCriteria'="News", 'TimeGroupingCriteria'="today" and 'TerminalCapability'="DVB-SH", (equal to the TerminalCapabilityCriteria set of values signalled in the 'TerminalCapability' element of DescriptorEntry  $X$ .)
- DescriptorEntry  $Y_1$  with 'ServiceCriteria'="TV2", 'GenreGroupingCriteria'="News", 'TimeGroupingCriteria'="today" and 'TerminalCapability'="DVB-H" (refers to the broadcast/unicast distribution system over which the current SGDD is retrieved)

in order to find the missing fragments for DescriptorEntry  $X$  with 'ServiceCriteria'="TV2", 'GenreGroupingCriteria'="News", 'TerminalCapabilityCriteria'="DVB-SH" and 'TimeGroupingCriteria'="tomorrow".

Note on forward compatibility:

As described above, if several criteria for grouping are present at the same time, all those grouping criteria apply to the set of Service Guide fragments in this Service Guide DescriptorEntry. The terminal would have to perform a logical 'AND' operation on the individual Grouping Criteria elements. Therefore addition of new Grouping criteria elements might not be handled well by BCAST 1.0 terminals.

So, in case forward compatibility with BCAST 1.0 terminals is intended, the TerminalCapabilityCriteria element SHALL NOT be used for the GroupingCriteria's of DescriptorEntries that announce service guide fragments that declare a service that is associated to the same broadcast/unicast distribution system over which the current SGDD is retrieved.

For example, suppose there are three underlying broadcast/unicast distribution systems ("DVB-H", "DVB-SH", "MBMS") required to consume the content and services declared by Service Guide fragments announced in three different Descriptor Entries respectively ("DE1", "DE2", "DE3"), that are all carried in a SGDD that is distributed over the DVB-H BDS. Then DescriptorEntry that declares Service Guide fragments that are associated with a service that is consumed via DVB-H has to omit the TerminalCapabilityCriteria element under the GroupingCriteria. The other two DescriptorEntries then declare the TerminalCapabilityCriteria ("DVB-SH", "MBMS") respectively.

#### 5.4.1.5.4 Availability of the SG fragments over broadcast and over interactive channel

In the 'DescriptorEntry' element of the 'SGDD' one can declare fragments to be available explicitly via broadcast and/or via interactive channel. The availability over broadcast channel is declared using the 'Transport' element of the 'DescriptorEntry' whereas the availability over interactive channel is declared using the 'AlternativeAccessURL'. The presence of these elements in the 'DescriptorEntry' SHALL be interpreted by the terminal as follows:

- If only 'Transport' element is instantiated the fragments declared in the corresponding 'DescriptorEntry' are available only over the broadcast channel according to the information in the 'Transport' element. In such a case the 'ServiceGuideDeliveryUnit' element represents a transport object in the broadcast file delivery session containing exactly those fragments that are listed within the 'ServiceGuideDeliveryUnit' element in question.
- If only 'AlternativeAccessURL' elements are instantiated the fragments declared in the corresponding 'DescriptorEntry' are available only over the interactive channel from the locations designated by the values of the 'AlternativeAccessURL' elements. Further, there does not necessarily exist any 'SGDUs' corresponding to the 'ServiceGuideDeliveryUnit' elements but these (ServiceGuideDeliveryUnit) elements merely act as lists of fragments that are available to be requested over the interactive channel from the address designated by the value of the 'AlternativeAccessURL' element.
- If both the 'Transport' element and at least one 'AlternativeAccessURL' element are instantiated the fragments declared in the corresponding 'DescriptorEntry' element are available over both the broadcast and the interactive channels.

### 5.4.2 Delivery over the Broadcast Channel

Over the Broadcast Channel, interface SG-5, the Service Guide is delivered using broadcast file delivery sessions. The network places the fragments of the Service Guide into one or more SGDUs and constructs one or more SGDDs to represent the contents of the Service Guide as well as the division of the fragments into the SGDUs. The SGDD(s) and the SGDU(s) are placed into file delivery session(s) to be transported as transport objects, TOs. While the SGDUs can be transported using one or more file delivery sessions, the SGDDs are provided using only one session, namely the Service Guide Announcement Channel as defined in section 5.4.1.5.1.

In the Service Guide Announcement Channel, the network SHALL use FLUTE [RFC 3926] as the broadcast delivery protocol and FDT Instances SHALL therefore be provided. In the Service Guide Delivery Channel, the network SHALL either use FLUTE (in which case FDT instances SHALL be provided) or ALC (in which case FDT Instances SHALL NOT be provided).

The following enhancements apply for the case when the file information is conveyed in the Service Guide or in a file delivery table:

- SG-D in BSD/A MAY apply the "Compact No-Code FEC scheme" [RFC 3695] (FEC Encoding ID 0, also known as "Null-FEC").
- SG-D in BSD/A MAY utilize the split-TOI scheme as specified in section 5.4.2.1.3 in conjunction with FLUTE, for signalling the identifier and version of any transported object (e.g. the Service Guide Delivery Unit or Service Guide Delivery Descriptor).
- SG-D in BSD/A MAY utilize the scheme as specified in section 5.4.2.1.3 in conjunction with FLUTE, for signalling the identifier and version of the Service Guide Delivery Unit.

In order for the terminals to distinguish the SGDDs and SGDUs from other transport objects the network SHALL set the 'Content-Type' attribute of the 'File' element in the FDT Instances

- to "application/vnd.oma.bcast.sgdd+xml" for SGDDs and
- to "application/vnd.oma.bcast.sgdu" for SGDUs.

As there is no signalling whether the network uses FDT Instances in the Service Guide delivery sessions other than the Service Guide Announcement Channel, the terminal



- SHALL assume that the Transport Object Identifier, TOI, zero is reserved for the FDT Instances.
- And the network SHALL not use the TOI zero for any types of files other than FDT Instance.

The network SHALL signal the Forward Error Correction, FEC, parameters for the transport objects in the Service Guide delivery sessions using one of the mechanisms defined in FLUTE [RFC 3926], and the terminal SHALL support all these mechanisms. When ALC/LCT is used for SGDU delivery, the FEC-encoding-ID SHALL be signaled in the LCT header codepoint field, and the EXT\_FTI extension header SHALL be used to signal other coding parameters.

#### 5.4.2.1 Signaling Changes in the Service Guide over Broadcast Channel

In the following, the way of signaling changes in Service Guide fragments is specified. The changes in the Service Guide are signalled through the change in the transmitted SGDUs which consequently cause a change in the transmitted SGDDs. Observing these changes, the terminal SHALL be able to determine the change. However, this specification does not specify the normative terminal behavior for this. Informative examples for four cases of localizing changes and achieving their discovery are outlined in section 5.5.1.1.

##### 5.4.2.1.1 Signalling Addition of a New SGDU

Upon addition of a new SGDU to be delivered on the Service Guide Delivery Channel, a new TOI is allocated for the delivery of the SGDU.

- If FLUTE is used on the Service Guide Delivery Channel, the allocated TOI SHALL be introduced in the FDT associated with Service Guide Delivery Channel.
- If ALC is used on the Service Guide Delivery Channel, it is assumed that the Terminal detects the change of TOI by observing the session.

Further, the allocated TOI SHALL be introduced in the SGDD on the Service Guide Announcement Channel. Consequently, the TOI of SGDD itself SHALL change. This SHALL be indicated through introducing the new TOI of the SGDD in the FDT associated with Service Guide Announcement Channel. While using FLUTE, the filenames associated with each Service Guide Delivery Descriptor within the FDT SHALL be set so that the terminal can use the FDT information to identify the transported Service Guide Delivery Descriptor, and its version.

- The 'Content-Location' attribute of the 'File' element within the FDT is used for this purpose. The Service Guide Delivery Descriptor is identified by its SGDDid (see section 5.4.1.5.2), which SHALL be used as a unique URI for the 'Content-Location' attribute.
- The version change of the Service Guide Delivery Descriptor is signalled based on the TOI and FDT Instance ID as specified in section 5.2.4 in [BCAST12-Distribution].

In addition, if Split-TOI scheme is used, the network MAY signal version relation between a removed SGDD and a newly inserted SGDD by allocating the TOI for the newly inserted SGDD, i.e. by taking the Most Significant Bits of the removed SGDD's TOI and changing only the Least Significant Bits of the TOI.

##### 5.4.2.1.2 Signalling Change in SGDUs

Upon change in SGDUs one or more SGDUs are inserted to or removed from the Service Guide Delivery Channel. The TOIs corresponding to the removed SGDUs SHALL be disassociated with the SGDDs which the SGDUs were associated with before, and the TOIs corresponding to the inserted SGDUs SHALL be signalled according to chapter 5.4.1.2.1.

- In addition, if Split-TOI scheme is used, the network MAY signal version relation between a removed SGDU and a newly inserted SGDU by allocating the TOI for the newly inserted SGDU, i.e. taking the Most Significant Bits of the removed SGDU's TOI and changing only the Least Significant Bits of the TOI.
- In addition, if Split-TOI scheme is used, the network MAY signal version relation between a removed SGDD and a newly inserted SGDD by allocating the TOI for the newly inserted SGDD by, i.e. taking the Most Significant Bits of the removed SGDD's TOI and changing only the Least Significant Bits of the TOI.

### 5.4.2.1.3 Split-TOI scheme

To provide a mechanism for terminals to easily track Service Guide updates when the Service Guide is delivered over broadcast channel using ALC or FLUTE, this section specifies a method to use the LCT TOI field of a transported SGDD/SGDU to indicate its identifier and its version, so that terminals can track the delivery of new versions of this SGDD/SGDU without parsing its payload.

When FLUTE is used, the Split TOI mechanism especially reduces the need to distribute FDT Instances functionally restricted to signal new versions of SGDDs/SGDUs (i.e. FDT Instances not describing new SGDDs/SGDUs, or not updating the “Expires” time for some SGDDs/SGDUs, etc.). Apart from the potential reduction of distributed FDT Instances, this mechanism still relies on the distribution of FDT Instances, and not does relieve the FD-C of its parsing obligations with regard to any received FDT Instances.

The LCT TOI field is  $32*O + 16*H$  bits in length where the Transport Object Identifier flag (O) length is 2 bits and the Half-word flag (H) length is 1 bit. The maximal length of the TOI is therefore 112 bits (i.e. 14 bytes). When a version identifier is assigned to a transported object through the LCT header, the TOI field is split into two parts: the first part (Most Significant Bits) called Object ID part is allocated to the identification of SGDU/SGDD, the second part (Less Significant Bits) called Version ID part is allocated to the version of the identified SGDU/SGDD. The terminal can track SGDU/SGDD updates based on the changes in the Version ID part of the TOI.

The receiver detects whether the TOI is split or not through in-band signalling as specified below:

- for an SGDD/SGDU transported in a FLUTE session, the indication of a split TOI SHALL be signalled in all FDT Instances declaring this SGDD/SGDU by the inclusion of “Version-ID-Length” attribute either in <FDT-Instance> top-level element or in the <File> element describing this SGDD/SGDU. The former case signals that the TOI of each SGDD/SGDU described in the FDT Instance is split, using “Version-ID-Length” specified at <FDT-Instance> element level unless overridden by “Version-ID-Length” provided at <File> element level.
- for an SGDU transported in an ALC session, the indication of split TOI SHALL be signalled in all SGDD structures describing this SGDU, by the inclusion of “versionIDLength” attribute in the element declaring the TOI for this SGDU (i.e. <ServiceGuideDeliveryUnit> element).

Whatever use is made of Split TOI scheme in an SG delivery session (such as TOIs split possibly with different Version ID lengths, mixed with TOIs not split), the uniqueness of each LCT TOI value generated in the session as well as the one-to-one mapping between TOIs and transport objects SHALL still apply.

Besides, for all SGDDs with a split TOI delivered in the same FLUTE session:

- there SHALL be a one-to-one mapping between SGDD ‘id’ (URI) and Object ID part in LCT TOI.
- SGDD “version” number SHALL be equal to Version ID part in LCT TOI.

The network MAY utilize the Split-TOI scheme for signalling the identifier and version of transported object, and terminals MAY be able to interpret the split TOI field in ALC header.

### 5.4.3 Delivery over the Interaction Channel

If a terminal has access to the Interaction Channel, then it SHALL support the mechanisms for accessing the Service Guide over Interaction Channel as defined by the provisions of this section.

The following gives a brief overview on the ways of requesting Service Guide over the Interaction Channel in the light of use cases enabled:

1. Terminal can request Service Guide fragments by their identifiers. This requires that the terminal knows the identifiers of the fragments prior to request. Consequently, the terminal can use this request to update the version of the fragments it already has or to retrieve fragments declared in the Service Guide Delivery Descriptor it had acquired earlier.
2. Terminal can request Service Guide Delivery Descriptors by their identifiers. This requires that the terminal knows the identifiers of the Service Guide Delivery Descriptors prior to request. Consequently, the terminal can use this request to update the version of the Service Guide Delivery Descriptors.

3. Terminal can specify within the request whether it wants responses as SGDDs or SGDU. This way the terminal can either get the declarations of fragments, the fragments itself or both.
4. Terminal can request Service Guide Delivery Descriptors by using the grouping criteria present in Service Guide Delivery Descriptors as request criteria. This way the terminal can request all Service Guide Delivery Descriptors fulfilling the given criteria. If the terminal also requested to have the fragments, it will receive the fragments that are available over Interaction Channel.
5. Terminal can request Service Guide fragments by using the attributes / elements present in Service Guide fragments as request criteria. This way the terminal can request all Service Guide fragments fulfilling the given criteria. If the terminal also requested to have the SGDDs, it will receive the SGDDs that declare the fragments.
6. Terminal can request Service Guide fragments and/or Service Guide Delivery Descriptors without any specific criteria or identifiers. This way terminal can request “any” view of Service Guide and let network to decide which set to provide to terminal.
7. Terminal can request all Service Guide fragments and/or Service Guide Delivery Descriptors. This way terminal can request the widest possible view to Service Guide.

In the use cases 3-7 the terminal should provide at least one BSMSSelector value as a part of the query to limit the response only to SG information that is associated with the terminal’s affiliated BSM(s).

#### 5.4.3.1 Rules applicable to all requests and responses

When requesting either Service Guide fragments or Service Guide Delivery Descriptors over Interaction Channel, the terminal and network SHALL comply with the following rules:

- The terminal SHALL originate requests. The network SHALL respond to requests.
- The request SHALL be made using ‘POST’ method of HTTP/1.1
- The parameters associated with the request SHALL be communicated as key-value pairs following the conventions defined in section 17.13 of [HTML4.01] for submitting HTML form data by the ‘POST’ method using the "application/x-www-form-urlencoded" encoding type. More specifically, once encoded as "application/x-www-form-urlencoded", the parameters to be passed from terminal to system SHALL be communicated in the ‘message-body’ of HTTP/1.1 ‘Request’ message as defined in section 5 of [RFC 2616].
  - Within a single request, the terminal MAY include multiple key-value pairs. As defined by [HTML4.01] these key-value pairs SHALL be delimited by an ‘&’.
  - Within the request, the terminal MAY specify the requested format of response.
    - When the terminal requests only Service Guide Delivery Descriptors, the terminal SHALL send “type=sgdd” as the first key-value-pair in the request.
    - When the terminal requests only Service Guide Delivery Units, the terminal SHALL send “type=sgdu” as the first key-value-pair in the request.
    - When the terminal requests Service Guide Delivery Descriptors and all Service Guide fragments declared within the Service Guide Delivery Descriptors that are available over Interaction Channel, the terminal SHALL send “type=sgdd+sgdu” as the first key-value-pair in the request.
    - When the terminal makes a specific request for SGDDs (section 5.4.3.3) but requests the result as SGDU, the server SHALL interpret this as a request to return all matching fragments associated with the SGDDs that the network would return if the network returned SGDDs as a result of the query.
    - When the terminal makes a specific request for SG fragments (section 5.4.3.4) but requests the result as SGDDs, the network SHALL interpret this as a request to return all currently valid and matching SGDDs associated with the requested SG fragments.

- When the terminal wants to minimize the information that are present in the server response, when no update occurred during a given period of time, it MAY insert in the ‘message-body’ of the HTTP/1.1 request the key equal to “lastResponseVersion”: the value of this key is then the last cached “lastResponseVersion” of the response which updates are requested by the terminal. The terminal can be also informed by the server until which time a requested given part of a Service Guide is expected to be up-to- date thanks to the use of the “ResponseValidity” element (see section 5.4.3.1.1).
- The following two notes give examples of contexts where the terminal can use “lastResponseVersion” and “ResponseValidity” information provided by the server

Note 1: for example, if the terminal did an unspecific request (See 5.4.3.2) for retrieving the service guide over the interaction channel, the server could, for instance, instantiate the SGResponse with SGDD(s) and SGDU according to a TimeGrouping criteria from present time to time Tx and inform the terminal that the (part of the) service guide is up-to-date up until Tx through the “ResponseValidity” element in the SGResponse. The server can also indicate the version of the current response in the “lastResponseVersion” attribute. The terminal can then repeat the same (unspecific) request after Tx to obtain a response containing the SGDD(s) and SGDU with the (part of the) service guide that is up to date until later time Ty.

Note 2: the terminal can also use this mechanism for the request of specific Service Guide Delivery Descriptors (see section 5.4.3.3) or SG fragments (see section 5.4.3.4) using various key and value pairs, such as the “globalServiceID” criterion. However, in the case when the terminal performed a request for specific SGDD(s) using “sgddID(s)” as the key(s), the terminal cannot assume it can repeat the initial request, i.e. using the same “sgddID(s)” values to obtain an update view of the target service guide parts. For instance, for a given fragment signalled at a given time by a given sgddID, the update of the fragment may be signalled at another time by a different sgddID.

- Furthermore, the terminal MAY announce a specific BCAST release number under which to interpret the request, using the key ‘bcastrelease’. In such case, the server SHALL provide a response according to the requested release number or SHALL announce the requested release number is not supported by using status code ‘012’ in the ‘status’ attribute of the “SGResponse” instance.
  - As of the present release of this specification, the value ‘1.0’ is defined. This is also the default value in case the ‘bcastrelease’ key is omitted.
- The response to the request SHALL be HTTP/1.1 response with status ‘200 OK’.
- The response SHALL have HTTP header field ‘Content-Type’ set to “application/octet-stream”
- The HTTP payload of response SHALL contain one instance of element “SGResponse”. See 5.4.3.1.1 for the definition of element “SGResponse”.
  - The element “SGResponse” carries the status code associated with the response, and the optional “lastResponseVersion” attribute.
  - If Service Guide Delivery Descriptors are returned with the response, those SHALL be carried within the element “SGResponse”.
  - If the optional ResponseValidity element is returned with the response, this SHALL be carried within the element “SGResponse”.
  - In case the ‘status’ attribute of the element “SGResponse” holds value ‘012’ or value ‘016’, the server SHALL neither instantiate the “ServiceGuideDeliveryDescriptor” nor provide an SGDU in the response.
- In the HTTP payload, the XML root closing tag of the element “SGResponse” MAY be immediately followed by one Service Guide Delivery Unit. If Service Guide fragments are returned with the response, those SHALL be encapsulated in that Service Guide Delivery Unit.

## 5.4.3.1.1 Definition of element "SGResponse"

Name	Type	Category	Cardinality	Description	Data Type
SGResponse	E			<p>Delivers the status of response to interactive Service Guide request. Also contains Service Guide Delivery Descriptors associated with the response.</p> <p>Contains the following attribute: status</p> <p>lastResponseVersion</p> <p>Contains the following elements: SupportedVersion ServiceGuideDeliveryDescriptor ResponseValidity PrivateExt</p>	
status	A	M	1	<p>Declares status of the interactive Service Guide delivery response using 'GlobalStatusCode' defined in the section 5.1.4 of [BCAST12-Services]. The following status codes SHALL NOT be used: 001-006, 009-011, 013-015, 019, 022, 024-027, 029-031</p>	unsigned Byte
lastResponse Version	A	O	0..1	<p>Signals the version of this response as built by the server. The purpose of this attribute is the following.</p> <p>When further request is sent by a terminal with the key "lastResponseVersion", the server behaviour is defined as follows:</p> <ul style="list-style-type: none"> <li>- the server SHOULD send a status code equal to "016" with no ServiceGuideDeliveryDescriptor element(s) and no ServiceGuide Delivery Unit if no update on the requested (part of the) service guide has been done compared to the version signalled by the "lastResponseVersion" value indicated in the request.</li> <li>- the server SHALL send a complete Response if any update on the requested (part of the) service guide has occurred (i.e. returns the same response as if the key "lastResponseVersion" was not present or not considered by the server).</li> </ul> <p>The behaviour of the server when the elements PrivateExt, and/or &lt;proprietary elements&gt; have been updated is not specified</p> <p>Note, that in both specified cases, the server MAY instantiate the element "ResponseValidity" e.g. to indicate a new "expirationTime" attribute and MAY instantiate the "lastResponseVersion" attribute.</p> <p>It is out of the scope of this specification to decide how this attribute is managed on the server side.</p>	unsignedInt

SupportedVersion	E1	M	0..N	This element is used in case the server responds with a 'GlobalStatusCode' of value '012' (unsupported version) to signal the BCAST release numbers supported by the server. If this element is not instantiated while the server responds with a 'GlobalStatusCode' of value '012', the terminal SHALL assume '1.0' is the supported release number.	string
ServiceGuideDeliveryDescriptor	E1	O	0..N	Service Guide Delivery Descriptor. See section 5.4.1.5.2.	complexType
ResponseValidity	E1	O	0..1	Signals information about the validity of the response. Contains the following attributes: expirationTime timeWindow	complexType
expirationTime	A	M	1	Its purpose is to indicate to the terminals that this received (part of the) service guide constituted by the SGDD(s) and/or SGDU provided in this response is scheduled to be at least up-to-date from the current response time up to the expirationTime' value. There is no assumption that the Service Guide element "PrivateExt", "<proprietary elements>" will be up-to-date until the expirationTime value. If "expirationTime" is present, a terminal that wants to track updates of this received (part of the) service guide, SHOULD not renew the request before the expirationTime is reached, without further instruction. If the "timeWindow" attribute is not instantiated, there is no assumption that this (part of the) service guide is up-to-date after the "expirationTime" time has been reached. This field is expressed as the first 32bits integer part of NTP time stamps.	unsignedInt

timeWindow	A	O	0..1	<p>Its purpose is to provide a mechanism that ensures distribution over time of a given update request sent from terminals, e.g. in order to avoid overload in network nodes or links.</p> <p>If "expirationTime" and "timeWindow" are present, a terminal that wants to track updates of this received (part of the) service guide SHOULD renew the request in the time interval [expirationTime, expirationTime+timeWindow].</p> <p>The exact time within the allowed time window SHALL be random with uniform probability.</p> <p>There is no assumption that this received (part of the) service guide is still up-to-date after the ("expirationTime"+"timeWindow") time has been reached.</p> <p>The unit is seconds (fractions can be expressed using data type Decimal."timeWindow" SHALL be a non-negative number.</p> <p>Note that if a large number of terminals have cached a server's response with the same expirationTime and without an appropriately-sized timeWindow instantiate', then there is a risk of network nodes/links congestion</p>	decimal
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.4.3.2 Unspecific request for retrieving service guide over Interaction Channel

As opposed to specific requests an unspecific request does not indicate which fragments (or descriptors) the terminal wants to receive. If the terminal supports unspecific requests for retrieving the service guide over Interaction Channel, the terminal SHALL request Service Guide fragments and Service Guide Delivery Descriptors over the Interaction Channel as follows:

- When terminal requests Service Provider's default view to Service Guide, the 'message-body' of HTTP/1.1 request message SHALL contain no data other than the optional key-value pairs defined in chapter 5.4.3.1: "Rules applicable to all requests and responses".
- The 'Request-URI' of HTTP POST request SHALL be set to the Service Guide entry point address (URL).
- The response of the request MAY contain Service Guide Delivery Descriptors, Service Guide fragments or both according with the "type=" key-value specified in the request.

When the response contains Service Guide fragments, the returned set of fragments SHALL be consistent as specified in 5.4.1.5.3.

### 5.4.3.3 Requests for specific Service Guide Delivery Descriptors

If the terminal supports requests for specific Service Guide Delivery Descriptors, the terminal SHALL request specific Service Guide Delivery Descriptors over the Interaction Channel as follows:

- When terminal requests individual Service Guide Delivery Descriptors by their identifiers, the 'message-body' of HTTP/1.1 request message SHALL contain one or several key-value pairs, using "sgddID" as the key and the attribute 'id' of the requested SGDD as the value. The network SHALL interpret this as a request to get the newest version of the requested SGDD.

- When terminal requests Service Guide Delivery Descriptors with criteria other than SGDD identifier, the ‘message-body’ of HTTP/1.1 request message SHALL contain one or several key-value pairs, using <key> as the key representing the criteria and the <value> as the value from the domain of the criteria. If several key-value pairs are given, they are combined as follows
  - key-value pairs having the same key are combined using OR logic, i.e. the expected reply are Service Guide Delivery Descriptors that satisfy at least one of the given criteria. The group of OR-combined keys is in the next step below treated as one entity.
  - Key-value pairs (or groups of pairs having the same key) having different keys are combined using AND logic, i.e. the expected reply are Service Guide Delivery Descriptors that satisfy all given criteria.
- Furthermore, terminal and server SHALL support the following assignments for the <key> and <value> fields:
  - <key> : “tgc-start”  
 <value> : Attribute ‘startTime’ associated with element ‘TimeGroupingCriteria’ used within SGDD  
 Meaning : Terminal requests SGDD grouped with ‘TimeGroupingCriteria’, whose ‘startTime’ equals to <value>.  
 Note : This <key> SHALL be used if and only if <key> “tgc-end” is used.  
 If this <key> is present, only one instance of this <key> SHALL be present.
  - <key> : “tgc-end”  
 <value> : Attribute ‘endTime’ associated with element ‘TimeGroupingCriteria’ used within SGDD  
 Meaning : Terminal requests SGDD grouped with ‘TimeGroupingCriteria’, whose ‘endTime’ equals to <value>.  
 Note : This <key> SHALL be used if and only if <key> “tgc-start” is used.  
 If this <key> is present, only one instance of this <key> SHALL be present.
  - <key> : “relative-tgc-start”  
 <value> : This field contains a time duration expressed in seconds.  
 Meaning : Terminal requests SGDD grouped with ‘TimeGroupingCriteria’, whose ‘startTime’ equals to the time when the request is received by the server increased by <value> (in form of the 32 bits integer part of the related NTP time stamp)  
 Note : This <key> SHALL be used only if <key> “relative-tgc-end” is used. When this key is not present and the key “relative-tgc-end” is present, the server SHALL interpret it as “relative-tgc-start=0”.  
 If this <key> is present, only one instance of this <key> SHALL be present.
  - <key> : “relative-tgc-end”  
 <value> : This field contains a time duration expressed in seconds.  
 Meaning : Terminal requests SGDD grouped with ‘TimeGroupingCriteria’, whose ‘endTime’ equals to the time when the request is received by the server increased by <value> (in form of the 32bits integer part of the related NTP time stamp)  
 Note : If this <key> is present, only one instance of this <key> SHALL be present.
  - <key> : “ggc”  
 <value> : Value of element ‘GenreGroupingCriteria’ used within SGDD.  
 Meaning : Terminal requests SGDD grouped with ‘GenreGroupingCriteria’, whose value equals to <value>.
  - <key> : “srvc”  
 <value> : Value of element ‘ServiceCriteria’ used within SGDD.  
 Meaning : Terminal requests SGDD grouped with ‘ServiceCriteria’, whose value equals to <value>.
  - <key> : “bsms”  
 <value> : Value of element ‘BSMSelector’ used within SGDD. The value is given as a string having the following syntax:



If the 'type' attribute of the 'BSMSelector' element in question equals "1":  
 "1;" <mobileCountryCode>;" < mobileNetworkCode>;" <networkSubsetCode> ";"  
 <networkSubsetCodeRangeStart> ";" < networkSubsetCodeRangeEnd > ";"  
 <serviceProviderCode> ";" <corporateCode> ";" <serviceProviderName>  
 If the 'type' attribute of the 'BSMSelector' element in question equal "2":  
 "2;" <nonSmartCardCode>

Individual values of the attributes (mobileCountryCode, mobileNetworkCode, etc.) SHALL be separated by single semicolons. A value of an attribute SHALL be encoded using digits and each attribute SHALL be immediately preceded and followed by a semicolon ';' even if the attribute value is empty, except the last attribute of the key which SHALL not be followed by a semicolon. Empty value is such that does not have any digits between the preceding and following semicolons. Empty value of an attribute means the attribute is not given.

The key SHALL begin with either digit "1" or digit "2" immediately followed by a semicolon. The individual attributes SHALL be ordered as listed above.

Meaning : Terminal requests SGDD grouped with 'BSMSelector', whose value equals to <value>.

Note : This <key> MAY be used when requesting SGDD(s) with a key other than "sgddID".

- <key> : "complete"
- <value> : "true"

Meaning : Terminal requests a complete set of SGDDs.

Note : This key SHOULD NOT be combined with criteria i.e. keys, restricting the set of SGDDs.

- <key> : "user"
- <value> : User id.

Meaning : Terminal requests a set of SGDDs personalized to the user associated with user identification of <value>.

- <key> : "SGEntryPointsOnly"
- <value> : "true"

Meaning : Terminal requests a "simplified" SGDD without any "DescriptorEntry". It is used to obtain declarations of SG entry points, associated to each related BSM. This request is sent only to interactive bootstrap URL, as discovered and explained in 6.2.

- <key> : "SGExclusivelyOverIC"
- <value> : "true"

Meaning : Terminal requests the SG parts that are exclusively delivered over the interaction channel. This can be typically used towards "complementary" or "superset" entry points over the interaction channel

- <key> : "AllSGOverIC"
- <value> : "true"

Meaning : Terminal requests the SG parts that are delivered over the interaction channel which can include fragments that are delivered over BC as well. This can be typically used towards "complementary" or "superset" entry points over the interaction channel.

- The 'Request-URI' of HTTP POST request SHALL be set to the Service Guide entry point address (URL).
- The response of the request MAY contain Service Guide Delivery Descriptors, Service Guide fragments or both.

#### 5.4.3.4 Requests for specific Service Guide fragments

If the terminal supports requests for specific Service Guide fragments, the terminal SHALL request specific Service Guide fragments over the Interaction Channel as follows:

- When terminal requests individual Service Guide fragments by their identifiers, the ‘message-body’ of HTTP/1.1 request message SHALL contain key-value pair, using "fragmentID" as the key and the attribute ‘id’ of the requested fragment as the value.
- When terminal requests Service Guide fragments associated to an SGDD, the ‘message-body’ of HTTP/1.1 request message SHALL contain key-value pair, using "sgddID" as the key and the attribute ‘id’ of the SGDD as the value.
- When terminal requests Service Guide fragments with criteria other than the fragment identifier or SGDD identifier, the ‘message-body’ of HTTP/1.1 request message SHALL contain key-value pairs, using <key> as the key representing the criteria and the <value> as the value from the domain of the criteria. These key-value pairs SHALL be delimited by a ‘&’. If several key-value pairs are given, they are combined as follows:
  - If there are several key-value pairs having the key “serviceType”, these are combined using AND logic, i.e. the expected reply are Service Guide fragments that satisfy all given criteria.
  - If there are several key-value pairs having the key “genre”, these are combined using AND logic, i.e. the expected reply are Service Guide fragments that satisfy all given criteria.
  - Otherwise, key-value pairs having the same key are combined using OR logic, i.e. the expected reply are Service Guide fragments that satisfy at least one of the given criteria. The group of OR-combined keys is in the next step below treated as one entity.
  - Key-value pairs (or groups of pairs having the same key) having different keys are combined using AND logic, i.e. the expected reply are Service Guide fragments that satisfy all given criteria.
- The response to a terminal request containing key-value pairs specifying the set of fragments the terminal expects to receive SHOULD contain all the fragments matching the given criteria and MAY include in addition fragments that do not match the given criteria. If the terminal request does not contain any key-value pairs having the keys “validFrom” or “validTo”, then the response SHALL contain only fragments that are currently valid.
- Furthermore, terminal and server SHALL support the following assignments for the <key> and <value> fields:
  - <key> : “globalServiceID”  
 <value> : Attribute ‘globalServiceID’ used within ‘Service’ fragment  
 Meaning : Terminal requests ‘Service’ fragments having ‘globalServiceID’ equal to <value> and Service Guide fragments associated with ‘Service’ fragments having ‘globalServiceID’ equal to <value>.
  - <key> : “globalContentID”  
 <value> : Attribute ‘globalContentID’ used within ‘Content’ fragment  
 Meaning : Terminal requests ‘Content’ fragments having ‘globalContentID’ equal to <value> and Service Guide fragments associated with ‘Content’ fragments having ‘globalContentID’ equal to <value>.
  - <key> : “globalServiceIDAll”  
 <value> : “true”  
 Meaning : Terminal requests all the ‘Service’ fragments and all the fragments that are associated with all the ‘Service’ fragments.
  - <key> : “globalContentIDAll”  
 <value> : “true”  
 Meaning : Terminal requests all the ‘Content’ fragments and all the fragments that are associated with all the ‘Content’ fragments.
  - <key> : “validFrom”  
 <value> : Attribute ‘validFrom’ used within Service Guide fragments  
 Meaning : Terminal requests Service Guide fragments whose attribute ‘validFrom’ is greater or equal to the <value>.
  - <key> : “validTo”  
 <value> : Attribute ‘validTo’ used within Service Guide fragments

- Meaning : Terminal requests Service Guide fragments whose attribute 'validTo' is less or equal to the <value>.
- <key> : "serviceType"  
 <value> : Attribute 'serviceType' used within 'Service' fragment  
 Meaning : Terminal requests 'Service' fragments having 'serviceType' equal to <value> and Service Guide fragments associated with 'Service' fragments having 'serviceType' equal to <value>.
  - <key> : "genre"  
 <value> : Element 'genre' used within Service Guide fragments  
 Meaning : Terminal requests Service Guide fragments having element 'genre' equal to <value> and Service Guide fragments associated with fragments having element 'genre' equal to <value>.
  - <key> : "fragmentEncoding"  
 <value> : Possible values listed in Table 3 under 'fragmentEncoding'.  
 Meaning : Terminal requests Service Guide fragments that are encoded with encoding scheme defined by <value>.
  - <key> : "fragmentType"  
 <value> : Possible values listed in Table 3 under 'fragmentType'.  
 Meaning : Terminal requests Service Guide fragments of that are of type <value>.
  - <key> : "all"  
 <value> : "true"  
 Meaning : Terminal requests all available Service Guide fragments.
  - <key> : "consistent"  
 <value> : "true"  
 Meaning : Terminal requests consistent set of Service Guide fragments.
  - <key> : "bsms"  
 <value> : Value of element 'BSMSelector' used within SGDD. The value is given as the following structure:  
  
 If 'type' equals "1":  
 "1;"<mobileCountryCode>;"< mobileNetworkCode>;"<networkSubsetCode>;"  
 <networkSubsetCodeRangeStart>;"< networkSubsetCodeRangeEnd >;"  
 <serviceProviderCode>;"<corporateCode>  
  
 If 'type' equals "2":  
 "2;"<nonSmartCardCode>  
  
 Individual values of the attributes (mobileCountryCode. mobileNetworkCode, etc.) SHALL be separated by single semicolons. A value of an attribute SHALL be encoded using digits and each attribute SHALL be immediately preceded and followed by a semicolon ';' even if the attribute value is empty, except the last attribute of the key which SHALL not be followed by a semicolon. Empty value is such that does not have any digits between the preceding and following semicolons. Empty value of an attribute means the value is not given.  
  
 The key SHALL begin with either digit "1" or digit "2" immediately followed by a semicolon. The individual attributes SHALL be ordered as listed above.
- Meaning : Terminal requests Service Guide fragments filtered with 'BSMSelector', whose value equals to <value>.
- Note : This <key> MAY be used when requesting Service Guide fragments with a key other than "fragmentID" or "sgddID".

- <key> : “modified-since”  
<value> : NTP time  
Meaning : Terminal requests a set of fragments modified since the specified point in time.
- <key> : “BroadcastAccess”  
<value> : Attribute ‘Type’ under element ‘BDSType’ in Access fragment  
Meaning : Terminal requests Service Guide fragments related to services receivable over the specified Broadcast access
- <key> : “UnicastAccess”  
<value> : Attribute ‘type’ under element ‘UnicastServiceDelivery’ in Access fragment  
Meaning : Terminal requests Service Guide fragments related to services receivable over the specified Unicast access
- <key> : “BDSCell”  
<value> : <BDSType>||“-” ||<BDSLocationID>

Possible values of <BDSType> are listed in CellTargetArea within ‘Service’ fragment. BDSLocationCellID is provided by terminal itself, and the structure depends on the value of <BDSType>. The acquisition of BDSLocationID is however out of the scope of this specification. BDSLocationID reuses the structure the value of CellArea as defined in each BCAST1.2 BDS adaptation specification.

Meaning : Terminal requests Service Guide fragments scoped by the given BDS cell. This parameter allows relating the Service Guide received over the Interaction channel to a BDS.

- <key> : “SGExclusivelyOverIC”  
<value> : “true”  
Meaning : Terminal requests the SG parts that are exclusively delivered over the interaction channel. This can be typically used towards “complementary” or “superset” entry points over the interaction channel
  - <key> : “AllSGOverIC”  
<value> : “true”  
Meaning : Terminal requests the SG parts that are delivered over the interaction channel which can include fragments that are delivered over BC as well. This can be typically used towards “complementary” or “superset” entry points over the interaction channel.
  - <key> : “startTime”  
<value> : Beginning of the period from which filtering is requested. This field contains the 32bits integer part of an NTP time stamp.  
Meaning : A fragment matches this criteria if it describes information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with the time interval specified by startTime/endTime
- Note : This <key> SHALL be used if and only if <key> “endTime” is used.  
If this <key> is present, only one instance of this <key> SHALL be present.
- <key> : “endTime”  
<value> : End of the period from which filtering is requested. This field contains the 32bits integer part of an NTP time stamp.  
Meaning : A fragment matches this criteria if it describes information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with the time interval specified by startTime/endTime
- Note : This <key> SHALL be used if and only if <key> “startTime” is used.  
If this <key> is present, only one instance of this <key> SHALL be present.

- <key> : “relativeStartTime”  
 <value> : Beginning of a relative period from which filtering is requested. This field contains a time duration expressed in seconds.  
 Meaning : A fragment matches this criteria if it describes information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with the time interval specified by  $(\text{relativeStartTime} + \text{requestTime}) / (\text{relativeEndTime} + \text{requestTime})$  where the requestTime is the time the request is received by the server (in form of the 32 bits integer part of the related NTP time)

Notes :

Unlike the key "startTime", this key represents a relative time from the time the request is received by the server.

This <key> SHALL be used only if <key> “relativeEndTime” is used.

If this <key> is present, only one instance of this <key> SHALL be present. When this key is not present and the key “relativeEndTime” is present, the server SHALL understand that the relative period starts from the time the request is received

- <key> : “relativeEndTime”  
 <value> : End of the period from which filtering is requested. This field contains a time duration expressed in seconds.  
 Meaning : A fragment matches this criteria if it describes information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with the time interval specified by  $(\text{relativeStartTime} + \text{requestTime}) / (\text{relativeEndTime} + \text{requestTime})$  where the requestTime is the time the request is received by the server (in form of the 32 bits integer part of the related NTP time)

Notes :

Unlike the key "endTime", this key represents a relative time from the time the request is received by the server.

If this <key> is present, only one instance of this <key> SHALL be present.

- <key> : “function”  
 <value> : Possible values are listed in Table 4 below.

Meaning : This key is used to confine the list of fragments associated with given service(s) (resp. with given content(s)). This key SHALL only be used in conjunction with other key(s) which lead to fragments that are associated with service(s) (resp. content(s)): i.e. the keys "globalServiceID", "globalContentID", "globalServiceIDAll", "globalContentIDAll", "genre", and "serviceType". The list of returned fragments is the subset of associated fragments that match the "function" <value>.

"function" Value	Meaning
“serviceAccess”	When this key value applies to associated fragments with a given service, it selects the Access fragments that reference directly to the service, the Schedule fragments that reference only to this particular service (i.e., do not reference to other fragments) and the Access fragments that reference to these Schedule fragments.
“contentAccess”	When this key value applies to associated fragments with a given content, its role is to select only the Schedule fragments (and related Access) that reference directly to this particular content
“servicePurchase”	When this key value applies to associated fragments with a given service, it selects only the provisioning information (‘PurchaseItem’, and related ‘PurchaseData’ fragments) related to the respective service i.e. the ‘PurchaseItem’ fragments that reference directly to the given service, and the set of ‘PurchaseData’ fragments that reference to these ‘PurchaseItem’ fragments

“contentPurchase”	When this key value applies to associated fragments with a given content, it selects the Schedule fragments that reference directly to this particular content and the provisioning information (‘PurchaseItem’, and related ‘PurchaseData’ fragments) related to the given Content, and Schedule fragments i.e. the ‘PurchaseItem’ fragments that reference directly the given Content fragment (resp. Schedule fragment), and the set of ‘PurchaseData’ fragments that reference to these ‘PurchaseItem’ fragments.
“serviceInteractivity”	When this key value applies to associated fragments with a given service, it selects only the ‘InteractivityData’ fragments that reference directly to the respective ‘Service’ fragment, and the fragments describing access and schedule information related to the respective ‘InteractivityData’ fragments
“contentInteractivity”	When this key value applies to associated fragments with a given content, it selects the Schedule fragments that reference directly to this particular content and the ‘InteractivityData’ fragments that reference directly to the given Content and Schedule fragment(s), and the fragments describing access and schedule information related to the respective ‘InteractivityData’ fragments.
“servicePreview”	When this key value applies to associated fragments with a given service, it selects the PreviewData fragments that are referenced to by this service and the Access fragments that are referenced to by these Preview Data fragments
“contentPreview”	When this key value applies to associated fragments with a given content, its role is to select the Schedule fragments that reference directly to this particular content and the PreviewData fragments that are referenced to by the given Content and Schedule fragment(s), and the Access fragments that are referenced to by these Preview Data fragments.

**Table 4: Semantics of possible values of the <key> "function"**

The following tables describe the role of the <key> “function”, depending on the possible joint use of the <key> “all” (defined later in this section):

	Associated fragments without function key filtering	function = contentAccess	function = contentPurchase	function = contentPreview	function = contentInteractivity
Fragments associated with Content + "all=false"	Schedule (+ related access) linked to the content	N/A (this value SHALL not be used in this case)	N/A (this value SHALL not be used in this case)	N/A (this value SHALL not be used in this case)	N/A (this value SHALL not be used in this case)
Fragments associated with Content + "all=true"	Schedule (+ related access) linked to the content Provisioning information linked to the content or to related schedules Preview linked to content (+ related access) or to related schedules (+ related access) Interactivity linked to content or related schedules (+ related access and	Schedules (+ related access) linked to the content	Schedule fragments linked to the content Provisioning information linked to the content and Provisioning information linked to the related Schedule fragments	Schedule fragments linked to the content Preview fragments (+ related access) linked to the content and Preview fragments (+ related access) linked to the related Schedule fragments	Schedule fragments linked to the content Interactivity fragments linked to the content and Interactivity fragments linked to the related Schedule fragments, and related access and schedule fragments to these Interactivity fragments)

	schedule)				
--	-----------	--	--	--	--

**Table 5: Effect of the <key> "function" on associated fragments with given content(s) depending on the presence of the <key> "all"**

	Associated fragments without function key filtering	function = serviceAccess	function = servicePurchase	function = servicePreview	function = serviceInteractivity
Fragments associated with Service + "all=false"	<p>Related contents fragments, and preview fragments linked to content fragments</p> <p>Interactivity fragments linked to the service (and related access and schedule)</p> <p>Access fragments directly linked to the service or linked to Schedule fragments that themselves only reference to the Service fragment</p>	<p>Schedule fragments directly linked to the service (but don't reference to any other fragments)</p> <p>Access fragments directly linked to the service and Access fragments linked to the related Schedule fragments</p>	N/A (this value SHALL not be used in this case)	N/A (this value SHALL not be used in this case)	Interactivity fragments linked to the service (and related access and schedule)
Fragments associated with Service + "all=true"	<p>Access, schedule linked to the service</p> <p>Provisioning information linked to the service</p> <p>Preview linked to the service and related access</p> <p>Interactivity linked to the service (and related access and schedule)</p> <p>Content fragments referencing to this service and all fragments associated with those contents</p>	Same as above	Provisioning information linked to the service	Preview fragments linked to the service (and related access fragments)	Same as above

**Table 6: Effect of the <key> "function" on associated fragments with given service(s) depending on the presence of the <key> "all"**

- The following applies for the selection of the target for the request:

- Upon the terminal requesting an individual Service Guide fragment by its identifier the terminal does this as follows: In case the terminal had acquired an SGDD that declared the fragment in question and the element 'AlternativeAccessURL', the 'Request-URI' of HTTP POST request SHALL be set to 'AlternativeAccessURL'.
- In any other case, the 'Request-URI' of HTTP POST request SHALL be set to the Service Guide entry point address (URL)
- In the response the requested Service Guide fragments SHALL be encapsulated in a SGDU as defined in section 5.4.1.3.
- If the request contained <key> "consistent" with <value> "true", the returned set of fragments SHALL be consistent as specified in 5.4.1.5.3.
- The following applies for requests for associated fragments:
  - If the request did not contain <key> "all" with <value> "true", then:
 

The term "fragments associated with a 'Content' fragment" specifies the fragments describing schedule information related to the respective 'Content' fragment and related access information.

The term "fragments associated with a 'Service' fragment" specifies the 'Content' fragments referencing that 'Service' fragment, the 'PreviewData' fragments related to those 'Content' fragments, the 'Access' fragments that reference directly to the 'Service' fragment and the 'Access' fragments that reference 'Schedule' fragments that themselves only reference to the 'Service' fragment (i.e. that doesn't reference either Contents or InteractivityData or PreviewData fragments). It also includes the fragments containing interactivity information related to the respective 'Service' fragment, and the fragments describing access and schedule information related to the respective 'InteractivityData' fragments,
  - If the request contained <key> "all" with <value> "true", then:
 

The term "fragments associated with a 'Content' fragment" specifies the fragments describing access, schedule, and provisioning information ('PurchaseItem' and related 'PurchaseData' fragments) related to the respective 'Content' fragment. It also provides the provisioning information related to the related 'Schedule' fragments, the fragments containing preview and interactivity information related to that 'Content' fragment and to the related schedule fragment(s), the fragments describing access and schedule information related to the 'InteractivityData' fragments, and the fragments describing access related to the 'PreviewData' fragments.

The term "fragments associated with a 'Service' fragment" specifies the fragments describing access, schedule, and provisioning information ('PurchaseItem' and related 'PurchaseData' fragments) related to the respective 'Service' fragment, the fragments containing preview and interactivity information related to that 'Service' fragment, the fragments describing access and schedule information related to the 'InteractivityData' fragments, the fragments describing access related to the 'PreviewData' fragments, the 'Content' fragments referencing that 'Service' fragment, and all the fragments associated with those 'Content' fragments.
  - Note that in requests for associated fragments the set of requested fragments can be confined by use of the <key> "fragmentType", and/or by the use of the <key> "function"..

### 5.4.3.5 Retrieving files referenced within Service Guide

When using the Interaction Channel for retrieving files referenced within Service Guide by URLs, the terminal and network SHALL comply with the following rules:

- The terminal SHALL support HTTP protocol for retrieving files referenced within Service Guide.
- The terminal SHALL originate requests. The network SHALL respond to requests.
- The request SHALL be made using 'GET' method of HTTP/1.1.
- The 'Request-URI' of HTTP GET SHALL be assigned with the URI to be retrieved.



- Files referenced within the Service Guide are external resources that might not have static content during the lifetime of the fragments that refer to those files. The terminal MAY cache such files. The terminal can detect updates of a cached file when re-issuing the corresponding HTTP request. Network and terminal MAY use HTTP/1.1 cache-control mechanisms to optimize update management. The network MAY signal updates to a file by updating the referencing URI.

## 5.5 Service Guide Update and Management

Over time, as services, contents and schedules come and go, the metadata associated with Service Guide changes. The terminal needs to be able identify the changes and to manage the updates of Service Guide and associated metadata accordingly. There are two ways the terminal SHALL support: 1) update and management of Service Guide over broadcast; and; 2) update and management of Service Guide over interaction. These ways differ in the specification of delivery method. However, in both of the cases the metadata management is done finally on the level of Service Guide fragments. Hence, the terminal SHALL be able to determine the identification, version and validity of each Service Guide fragment using the respective fields in the Service Guide fragment itself. Further, the terminal SHALL be able to interpret the following cases:

- Fragment to be processed has fragment id that has not been previously received. This means that the terminal has received a new piece of metadata.
- Fragment to be processed has same fragment id and same version than has been previously received. This means that no change is implied. The metadata that was already received is still valid.
- Fragment to be processed has same fragment id but higher version than has been previously received. This means that the metadata related to the fragment in question needs to be updated. If 'validFrom' attribute of the new fragment is not present or indicates some time in past, the terminal SHALL replace the previous metadata with the newly received metadata immediately. If 'validFrom' attribute of the new fragment is present and indicates some time in future, the terminal SHALL store the new fragment and keep using the existing one until the new one gets valid.
- Fragment to be processed has same fragment id but lower version than has been previously received. This means that the previously received metadata related to the fragment in question is more recent than the metadata in the fragment that was just received. Terminal SHALL discard the new fragment.
- The terminal SHALL handle the possible wrap-around of the version.

Further, overall validity of a Service Guide fragment MAY be expressed with OPTIONAL attributes 'validFrom' and 'validTo'. The terminal SHALL support the interpretation of these attributes in determining whether a given Service Guide fragment is valid on a given moment of time: If the current time is within the time span defined by 'validFrom' and 'validTo' the terminal SHALL consider that the metadata is a valid member of the set of fragments comprising the SG. Outside of the specified validity, the fragment SHALL NOT be considered part of the currently applicable SG (e.g. with respect to consistency), and the fragment information SHOULD NOT be used by the terminal, or rendered to the user at this point of time.

If the set of fragments belonging to the Service Guide are announced using the mechanism defined in section 5.4.1.1, then the terminal MAY assume from the absence of any fragment in the updated version of the SGDDs that the validity of the fragment has ended.

### 5.5.1 Service Guide Update and Management using the Broadcast Channel

Over the broadcast channel the transmitted Service Guide can be large and consist of several delivery sessions carrying several parts of the Service Guide. Thus, as the total size of Service Guide can be large – both in terms of data amount per fragment as well as in terms of fragments – it is useful to localize the changes to parts of Service Guide. This allows a terminal to identify as narrow changes as possible. Further it allows the terminal to receive the updates and manage data only for the relevant part(s) of Service Guide.

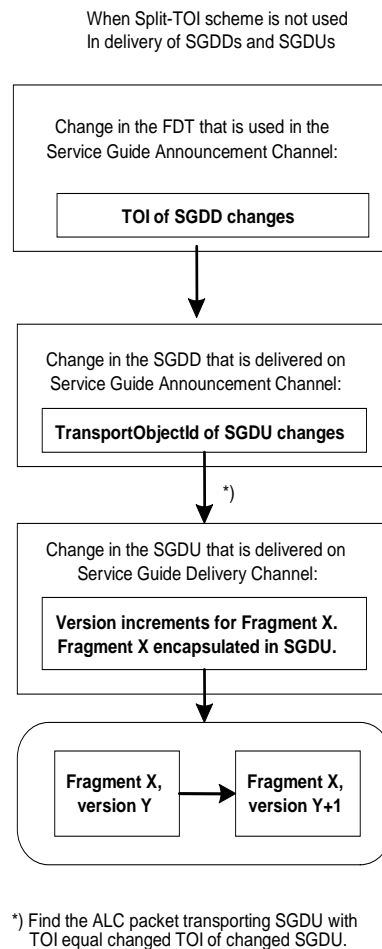
On the other hand, assuming that the terminal has received the Service Guide or a part of Service Guide, the structure of the Service Guide makes it possible for the terminal to determine quickly whether there is relevant information available in the Service Guide, which the terminal does not possess yet. Since the content of the Service Guide can and do change over time,

and one aims to minimize the time needed by the terminals to update their view of the Service Guide, one wants to have a mechanism for declaring the contents of the Service Guide without forcing the terminal to traverse through the entire set of fragments of the Service Guide.

The methods for localized changes and their discovery, and, for determining the well-defined subsets of the Service Guide are specified in the following sections.

### 5.5.1.1 Localized changes and their discovery (Informative)

The flow of changes and how they propagate in the Service Guide is illustrated in the Figure 3-Figure 6 below. This way even the smallest changes in the Service Guide (i.e. change in individual fragments) can be accurately pointed out.

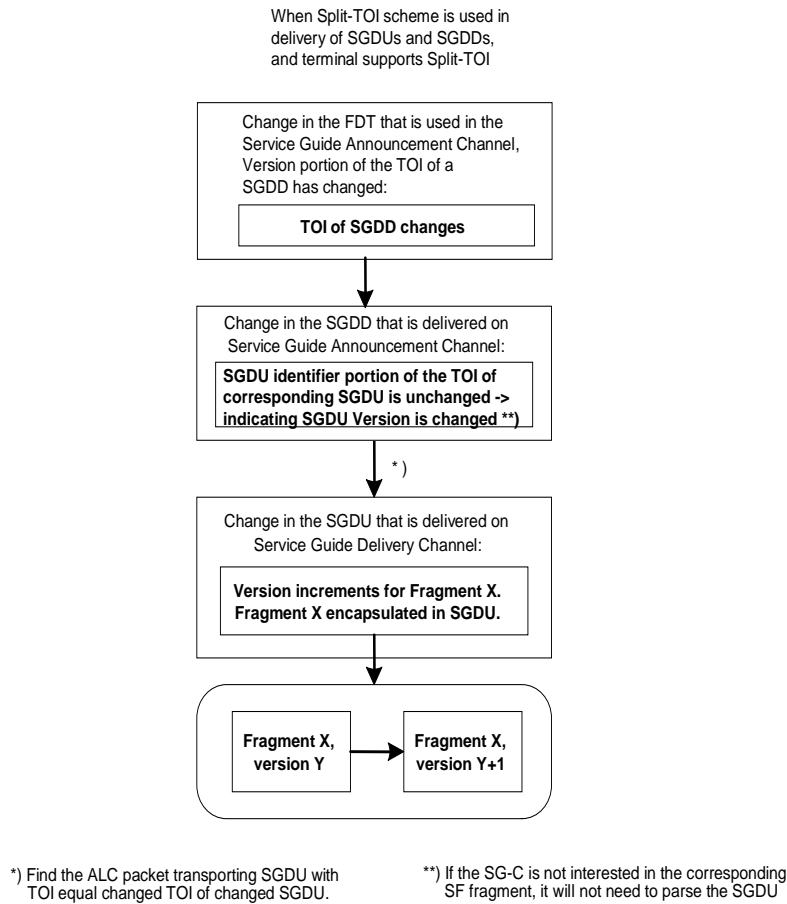


**Figure 3: SG fragment change and its propagation on the different levels of Service Guide, when Split-TOI is not used neither for SGDU nor for SGDD**

(Note: when FLUTE is used as the delivery of SGDUs, the FDT of the session carrying the SGDUs is also updated when TOI of the SGDU is changed.)

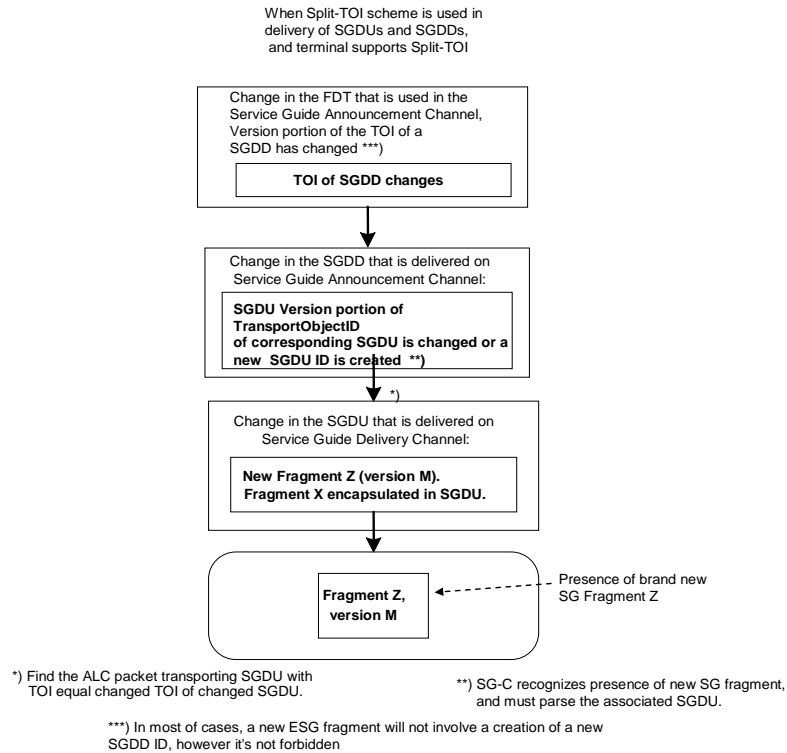
This flowchart in Figure 3 assumes that Split-TOI scheme is not used in the delivery of SGDDs/SGDUs. Here, the change of Fragment X version Y to Fragment X version Y+1 causes a change in the respective SGDU, whose TOI changes on the Service Guide Delivery Channel. Consequently, it causes the declaration of the SGDU in the respective SGDD to change. Further, as the SGDD changes, so does its Transport Object Identifier on the Service Guide Announcement Channel. Finally the SG-C notices this change through observing the change in the FDT table available on the Service Guide Announcement

Channel. Correspondingly, if the original corresponding fragment was of interest to the terminal, it will store and parse the associated SGDU. However, if that original fragment was not of interest to the SG-C, then the fragment version update would similarly be of no interest, and the SG-C will disregard the associated SGDU.



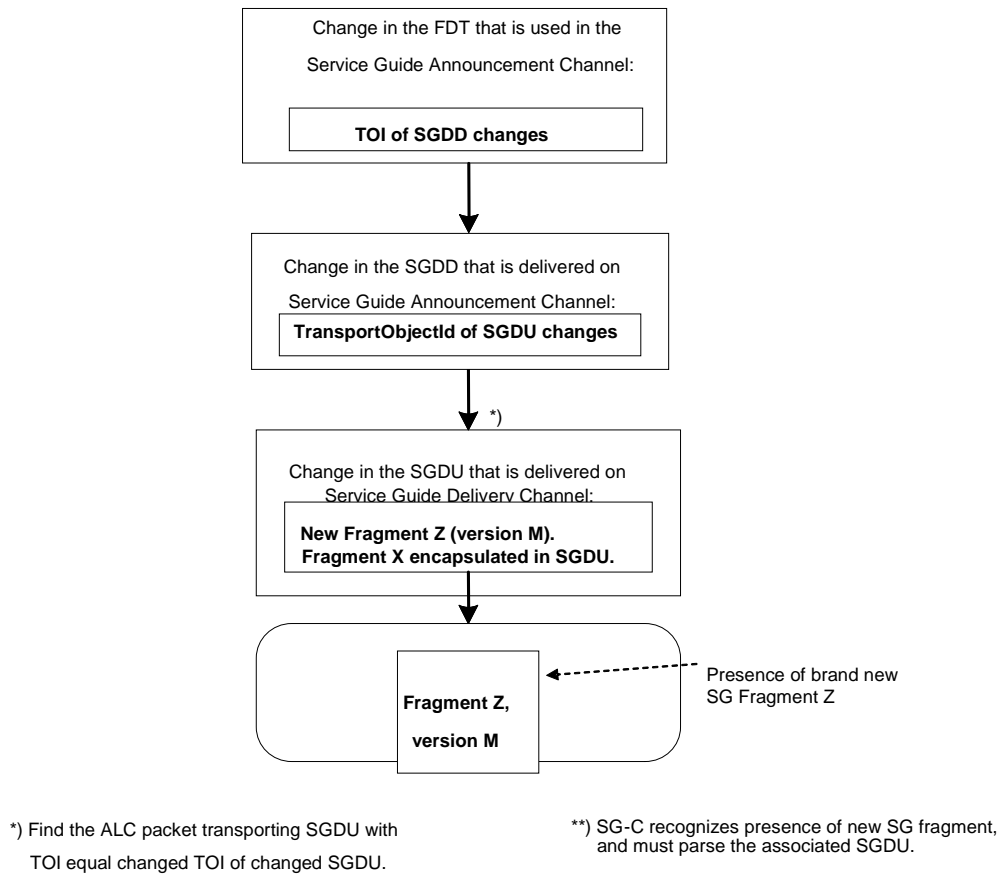
**Figure 4: Fragment version change and its detection at ALC packet header level of SGDU/SGDD delivery, when Split-TOI is used**

The flowchart in Figure 4 assumes that Split-TOI scheme is used in the delivery of SGDUs/SGDDs. Here, the change of Fragment X version Y to Fragment X version Y+1 causes a change in the respective SGDU, whose Version portion of the TOI is changed on the Service Guide Delivery Channel. Consequently, it causes the declaration of the SGDU in the respective SGDD to change. Further, as the SGDD changes, so does its Transport Object Identifier on the Service Guide Announcement Channel, and particularly the Version portion of the TOI. The SG-C notices this change through observing the change in the FDT available on the Service Guide Announcement Channel. In addition, the SG-C is able to detect the indicated SG fragment change as a version update. Correspondingly, if the original corresponding fragment was of interest to the terminal, it will store and parse the associated SGDU. However, if that original fragment was not of interest to the SG-C, then the fragment version update would similarly be of no interest, and the SG-C will disregard the associated SGDU.



**Figure 5: New fragment introduced and its detection at ALC packet header level of SGDU delivery, when Split-TOI is used**

The flowchart in Figure 5 assumes that Split-TOI scheme is used in the delivery of SGDUs. Here, the presence of a brand new SG fragment Z causes a change in the respective SGDU, whose Version portion of the TOI is changed on the Service Guide Delivery Channel, or causes a new SGDU identifier portion to be created. Consequently, it causes the declaration of the SGDU in the respective SGDD to change. Further, as the SGDD changes, so does its Transport Object Identifier on the Service Guide Announcement Channel, and particularly the Version portion of the TOI. The SG-C notices this change through observing the change in the FDT available on the Service Guide Announcement Channel. Correspondingly, to find out about this new fragment, the SG-C will store and parse the associated SGDU.



**Figure 6: New fragment introduced and its propagation on the different levels of Service Guide, when Split-TOI is not used**

This flowchart in Figure 6 assumes that Split-TOI scheme is not used in the delivery of SGDDs/SGDUs. Here, the presence of a brand new SG fragment Z causes a change in the respective SGDU, whose Object identifier portion of the TOI is changed on the Service Guide Delivery Channel. Consequently, it causes the declaration of the SGDU in the respective SGDD to change. Further, as the SGDD changes, so does its Transport Object Identifier on the Service Guide Announcement Channel. Finally the SG-C notices this change through observing the change in the FDT available on the Service Guide Announcement Channel. As the SG-C in the terminal notices the change in either of these ways, following the chain of propagation allows the SG-C to accurately locate the changed parts of the Service Guide and only to focus on receiving those.

### 5.5.1.2 Enabling Terminal to determine Service Guide completeness

At each given time, the terminal is proposed a set of SGDD over the Service Guide Announcement Channel, in the form of Transport Objects.

In addition to the construction rules defined in the previous section, in order for the terminal to determine it is receiving a consistent set of the Service Guide, the following signalling in the FLUTE FDT is specified. A new attribute ‘FullFDT’ is added to the ‘FDT-Instance’ element of the FDT. This attribute signals a complete description of all the Transport Objects that are currently scheduled for transmission in the corresponding FLUTE session.

The XML syntax of the ‘FullFDT’ attribute is the following:

```
<attribute name="FullFDT" type="boolean" use="optional" default="false" />
```

When the 'FullFDT' attribute is set to "true" in the FDT instance of a given FLUTE session, this means that all the Transport Objects that are scheduled in the said FLUTE session are described. When the 'FullFDT' attribute is left to false, the terminal cannot make any assumptions about the description. To enable SG-C in the terminal to determine any subset of a Service Guide the SG-D SHALL comply with the following rules:

- In the context of the FLUTE session providing the Service Guide Announcement Channel, the 'FullFDT' attribute SHALL always be set to "true".
- The set of fragments declared by the SGDDs SHALL be exhaustive, i.e., each fragment in the SG has to be declared at least in one SGDD

In addition, should several FDT instances be available at the same time, the terminal SHALL only consider the FDT Instance with the highest value of the FDT Instance ID taking into account the possible wrap around of the FDT Instance ID.

### 5.5.1.3 Terminal behaviour determining the Service Guide completeness (Informative)

The following state diagram Figure 7 below defines the deduction algorithm for Service Guide completeness when FLUTE is used for SGDD delivery. In principle, the exhaustive listing of fragments makes it possible to determine precisely the fragments that are needed and their location in the Service Guide delivery.

Explanation of variables used in the algorithm:

$L_D$	Set of SGDDs representing a subset of Service Guide.
$L_P$	Previously stored set of SGDDs.
$L_R$	Set of SGDDs to be received.
$F_R$	Set of fragments to be received
$F_D$	Set of fragments to be deleted
$C_R$	Set of SGDUs delivering the fragments in $F_R$

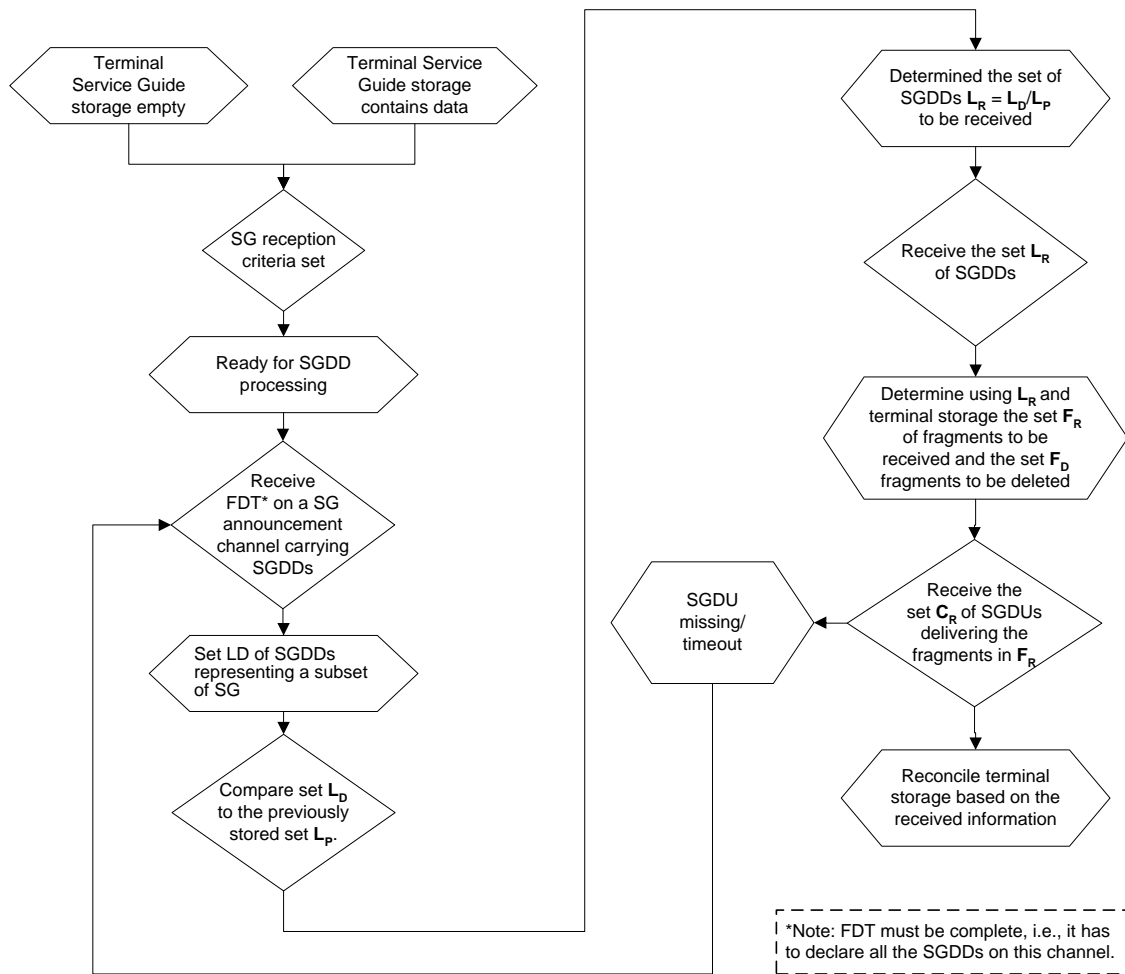


Figure 7: Algorithm for determining a subset of Service Guide

## 5.5.2 Service Guide update and management using the Interaction Channel

The Service Guide update and management over the Interaction Channel is enabled in two ways: In a terminal-based way and in a system-based way. The terminal-based way and system-based way can be applied mutually complementing each other.

In the terminal-based way the terminal keeps the state of version and validity of acquired Service Guide fragments and Service Guide Delivery Descriptors. Based on that information and the information available via Service Guide Delivery Descriptors possibly made available over Broadcast Channel the terminal detects the changes and reacquires the necessary fragments. The terminal can also poll for changes by trying to reacquire the already acquired Service Guide fragments and/or SGDDs over the Interaction Channel and deduce the changes that way. The delivery of Service Guide over Interactive Channel is normatively specified in section 5.4.3.

In the system-based way the terminal requests the system to keep the terminal updated upon changes on the requested Service Guide fragments and/or on Service Guide Delivery Descriptors. If the terminal supports OMA PUSH the terminal SHALL support the system-based way of update and management using the Interaction Channel as follows:

- Upon terminal requesting Service Guide Delivery Descriptors or Service Guide fragments through means as specified in section 5.4.3., the ‘message-body’ of HTTP/1.1 request SHALL be suffixed with string “&keep-updated=true” meaning that terminal requests the system to keep the terminal updated on changes to requested Service Guide Delivery Descriptors or Service Guide fragments.

- Upon changes or updates to the Service Guide fragments and/or Service Guide Delivery Descriptors for which the terminal has requested system to keep the terminal updated the following happens. The system SHOULD send the updated Service Guide fragments and/or Service Guide Delivery Descriptors to the terminal using OMA PUSH.
  - For Service Guide fragments, the MIME type “application/vnd.oma.bcast.sgdu” SHALL be used to identify that the PUSH message carries an SGDU containing the fragments.
  - For Service Guide Delivery Descriptor, the MIME type “application/vnd.oma.bcast.sgdd+xml” SHALL be used to identify that the PUSH message carries an SGDD. Each SGDD will be delivered as a separate PUSH message.

### 5.5.3 Service Guide Update and Management Cases for Hybrid Broadcast/Interactive Scenario (Informative)

When Service Guide is delivered in a hybrid broadcast/unicast environment, three cases for Service Guide update and management can be envisioned. These cases are described in the following subsections.

#### 5.5.3.1 Service Guide partly Distributed over Broadcast Channel and partly Distributed over Interaction Channel

The complete Service Guide consists of fragments that are delivered over both the broadcast channel and the interaction channel. Some of the fragments are only available over the broadcast channel and some of the fragments are only available over the interaction channel. See section I.2 for an explanation of the relationship between the services and the Service Guide fragments in this case.

For discovering and localizing changes made to the fragments delivered over the broadcast channel, the methods explained in section 5.5.1 can be used. Correspondingly, for discovering and localizing changes made to the fragments delivered over the interaction channel, the terminal-based way or the system-based as described in section 5.5.2 can be used. In addition, the interaction channel Service Guide update and management can also be eased by using the “lastResponseVersion” and/or “ResponseValidity” information provided by the server as described in section 5.4.3.1.

The server needs to take into consideration that terminals might not have access to both the broadcast channel and interaction channel when determining which access method(s) to use for delivering each individual fragment. In other words, the set of fragments delivered over the broadcast channel should be consistent as described in section 5.4.1.5.3, and the same applies for the whole set of fragments available over the interaction channel.

#### 5.5.3.2 Service Guide completely Distributed over both Broadcast Channel and Interaction Channel

The Service Guide is completely broadcasted, and the same version of the Service Guide is also alternatively available over the interaction channel. In this case, the terminal can use the broadcasted Service Guide to detect changes or updates in the Service Guide, as described in section 5.5.1. Alternatively, Service Guide update and management is done according to the terminal-based way or the system-based, as described in section 5.5.2 for the fragments retrieved over the interaction channel. For the terminal-based way, the updates can be eased by using the “lastResponseVersion” and/or “ResponseValidity” information provided by the server as described in section 5.4.3.1.

#### 5.5.3.3 Service Guide Distributed completely over Interaction Channel

The Service Guide is completely distributed over the interaction channel only and describes services available over the broadcast channel or services available over both the broadcast channel and interaction channel. In this case, the Service Guide update and management is done according to the terminal-based way or the system-based as described in section 5.5.2. For the terminal-based way, the updates can be eased by using the “lastResponseVersion” and/or “ResponseValidity” information provided by the server as described in section 5.4.3.1



## 5.6 Service Guide Backend Interfaces

The following sections specify interfaces between logical BCAST “backend” entities. The specification is applicable if the interfaces are exposed in a BCAST implementation. If a BCAST implementation does not expose the interfaces, i.e. they are implementation internal, they may be realized using protocols and methods not specified here. If a BCAST implementation does expose the interfaces, the network SHALL support the Service Guide Backend Interfaces syntax as defined by XML Schema in [BCAST12-Schema-sg-be].

### 5.6.1 Service Guide management in the backend

The server side clients of the BCAST represented by the logical entity ‘Content Creation’, CC, are regarded as providing information about services and/or individual pieces of contents for the BCAST system, which are in turn supplied to the users consuming the services. This exchange of information is performed over the interface SG-1 between the ‘Service Guide Content Creation Source’ entity, SGCCS, of the CC and the ‘Service Guide Application Source’ entity, SGAS, of the BSA. The consumers of the services get this information in the form of the Service Guide, SG, via the head-end interfaces SG-5 and SG-6. These interfaces are not provided directly by SGAS, but instead by the ‘Service Guide Distribution’ entity, SG-D, residing in the BSD/A. The SGAS supplies the information it receives from CC first to the ‘Service Guide Generation’ entity, SG-G, also residing in BSD/A, over the interface SG-2. The SG-D then receives the information from the SG-G via an internal interface in BSD/A. In addition to receiving information from SGAS, the SG-G also aggregates Service Provisioning information provided by the ‘Service Guide Subscription Source’ entity, SGSS, of the BSM. This exchange is performed over the interface SG-4.

This arrangement makes it possible for the BSD/A

- to act as a central control point of the program/content control on behalf of the CC, BSA and BSM by collecting information from BSA, BSM and internally from the different entities in BSD/A,
- to act only as an aggregator of the program/content specific portion of the Service Guide in the form of Service Guide fragments to be delivered for the terminals through head-end interfaces SG-5 and SG-6 whereas the BSA is given the control of creating the content-specific Service Guide fragments on behalf of the Content Creation entities or
- to act in some combined mode of the two.

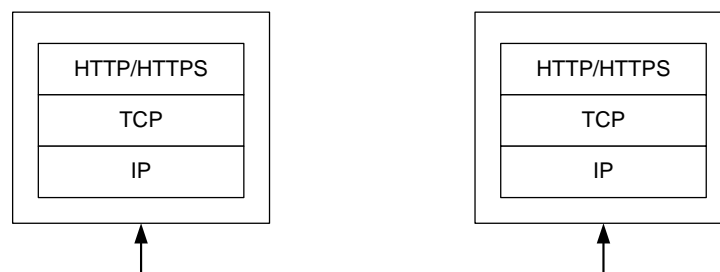
The arrangements in practise between the BSA, BSM and BSD/A with respect to Service Guide management are, however, out of the scope of this specification.

### 5.6.2 Service Guide contents exchange

This specification defines only the interface SG-4 while leaving interfaces SG-1 and SG-2 undefined.

#### 5.6.2.1 Protocol Stacks

The following protocol stack SHALL be used for exchanging messages between the backend SG entities. HTTP or HTTPS that SHALL be based on SSL 3.0 [SSL30] and TLS 1.0 [RFC2246] over TCP/IP SHALL be used for the delivery of messages.



Messages to and from CC, BSA, BSD/A or BSM are transported using HTTP by placing both the requests and the responses addressed to CC, BSA, BSD/A or BSM into the payload of the HTTP messages. The requests SHOULD be transported using

HTTP POST and the responses SHOULD be transported using the HTTP responses corresponding to the HTTP POST requests. The syntax for the requests SHOULD be as follows:

- POST <host>/oma/bcast1.2/sg HTTP/1.1\r\n<request>

where the <host> denotes the part of the URI representing the address of the host and the <request> denotes the XML element providing the request parameters.

Both the HTTP POST message and the corresponding HTTP response MAY also contain the following HTTP header fields:

- 'Content-Length',
- 'Content-Type' which if used SHALL be set to "text/xml" and
- 'Host' in case the 'Request-URI' is not in the absolute form specified in [RFC 2616].

### 5.6.2.2 Service Guide content delivery

This section specifies the service guide delivery message from SGSS to SG-G via SG-4 interface.

#### 5.6.2.2.1 Delivery Message

The following is the delivery message sent from SGSS to SG-G over the interface SG-4. In order to reflect any possible grouping of SG in the head-end interfaces SG-5 and SG-6, SGSS can inform SG-D via SG-G of the grouping by associating each 'SGData' element with one or more 'BSMSelector' elements and/or 'bsmSelectorID' attributes in the 'SGDelivery' request.

Name	Type	Category	Cardinality	Description	Data Type
SGDelivery	E			Specifies the delivery message of Service Guide data over interface SG-4 which is used for generating Service Guide in SG-G. Contains the following elements: BSMSelector BSMSelectorID SGData PrivateExt	
BSMSelector	E1	M	0..N	This element provides the details on the visibility of the enclosed 'SGData'. All the 'BSMSelectorID' values used in the request SHALL have one and only one of these 'BSMSelector' elements instantiated with matching identifier. Element 'BSMSelector' is specified in section 5.4.1.5.2.	complexType
BSMSelectorID	E1	M	0..N	This element represents constraints on the visibility of the all enclosed 'SGData' elements. This identifier corresponds to the 'id' attribute of the 'BSMSelector' element. See the corresponding description in the 'SGDD' for more details.	anyURI

SGData	E1	M	1..N	<p>Contains source information to be included into the Service Guide. It is RECOMMENDED that the information is delivered in the form of BCAST Service Guide fragments.</p> <p>Contains the following attributes:</p> <ul style="list-style-type: none"> <li>id</li> <li>transportID</li> <li>version</li> <li>validFrom</li> <li>validTo</li> <li>encoding</li> <li>type</li> </ul> <p>Contains the following element:</p> <ul style="list-style-type: none"> <li>Body</li> </ul>	
id	A	M	0..1	Identifier of the data enclosed in element 'Body'. See also the description of the 'id' attribute in the 'SGDeliveryRes' message.	anyURI
transportID	A	O	0..1	Transport identifier of the data enclosed in element 'Body'.	unsignedLong
version	A	M	1	Version of the data enclosed in element 'Body'.	unsignedInt
validFrom	A	M	1	Start time of the validity of the data enclosed in element 'Body'.	unsignedInt
validTo	A	M	1	End time of the validity of the data enclosed in element 'Body'.	unsignedInt
encoding	A	M	1	<p>Fragment encoding type of the data enclosed in element 'Body'.</p> <ul style="list-style-type: none"> <li>0 – XML encoded OMA BCAST Service Guide fragment</li> <li>1 –127 Reserved for future use</li> <li>128-255 Reserved for proprietary use</li> </ul>	unsignedByte
type	A	M	0..1	<p>Fragment type of the data enclosed in element 'Body'. This SHALL be present and set to one of the values listed below if the 'Body' contains a Service Guide XML fragment and the value of 'encoding' is set to '0'.</p> <ul style="list-style-type: none"> <li>5 – PurchaseItem Fragment</li> <li>6 – PurchaseData Fragment</li> <li>7 – PurchaseChannel Fragment</li> </ul>	unsignedByte
Body	E2	M	1	Contains the delivered Service Guide data. The value SHALL be an instance of 'PurchaseItem', 'PurchaseData' or 'PurchaseChannel' element as specified in 5.1.2.6, 5.1.2.7 and 5.1.2.8.	complexType
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.6.2.2.2 Response Message

The following is the response message for 'SGDelivery' request to be used over interface SG-4.

Name	Type	Category	Cardinality	Description	Data Type
SGDeliveryRes	E			Specifies the response message to be used in the response to 'SGDelivery' HTTP request. Contains the following elements: SGDataResult PrivateExt	
SGDataResult	E1	M	0..N	Represents the status for each of the 'SGData' elements of the corresponding HTTP request. If there are more than one 'SGData' element present in the HTTP request, these elements SHALL correspond to the 'SGData' elements of the HTTP request in the same order as in the HTTP request. Contains the following attributes: id statusCode	
id	A	M	1	Identifier of the Service Guide data contained in the 'SGData' element of the HTTP request. The assignment of this identifier is performed as follows: CASE 1: The value of the 'id' attribute in the 'SGData' element of the 'SGDelivery' HTTP request equals 'oma-bcast-unknown'. In this case the identifier of the fragment SHALL be assigned by the issuer of the response message and the value of the identifier SHALL be set as the value of the 'id' in this response. CASE 2: The value of the 'id' attribute in the 'SGData' element of the HTTP request does not equal 'oma-bcast-unknown'. In this case the identifier of the fragment SHALL be interpreted as assigned by the issuer of the 'SGDelivery' HTTP request and the issuer of the response SHALL simply copy the value of the 'id' attribute of the request into the 'id' attribute of the response. The issuer of the 'SGDelivery' HTTP request SHALL use this returned identifier in any future 'SGDelivery' HTTP requests representing updates on the information in the 'SGData' element in question.	anyURI
statusCode	A	M	1	Indicates the outcome of the processing of the 'SGData' element with matching value of 'id' attribute. The value SHALL be set as specified in [BCAST12-Services] for global status codes.	unsignedByte
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

### 5.6.2.3 Service Protection Information Requests

In order to be able to fill in information about possible service and content protection into ‘Service’ or ‘Content’ fragments, the SG-G of BSD/A needs to be able to request the corresponding parameters from the SGSS of BSM representing in turn access to the ‘SP Management’ of BSM. This request from SG-G to SGSS over SG-4 SHALL be an HTTP POST having an instantiation of the following ‘SGSPServiceInfo’ element as the payload.

Name	Type	Category	Cardinality	Description	Data Type
SGSPServiceInfo	E			Specifies the message for SG-G to request the value of the ‘baseCID’ and ‘ProtectionKeyID’ to be used in a ‘Service’ or ‘Content’ fragment and to query if a particular service should be signalled as protected or not from the SGSS. Contains the following attributes: globalServiceID globalContentID Contains the following element: PrivateExt	
globalServiceID	A	M	0..1	Globally unique identifier of the service in question. This is the value of the element ‘GlobalServiceID’ in the ‘Service’ fragment representing the service in question.	anyURI
globalContentID	A	M	0..1	Globally unique content identifier of the piece of content in question. This is the value of the element ‘GlobalContentID’ in the ‘Content’ fragment representing the piece of content in question.	anyURI
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

The following is the response message for the ‘SGSPServiceInfo’ to be used over the interface SG-4. The response from SGSS of BSM to SG-G of BSD/A SHALL be an instantiation of the following ‘SGSPServiceInfoRes’ message placed into the payload of the corresponding HTTP response.

Name	Type	Category	Cardinality	Description	Data Type
SGSPServiceInfoRes	E			Specifies the response message for a ‘SGSPServiceInfo’. Contains the following attributes: statusCode baseCID Contains the following elements: ProtectionKeyID PrivateExt	
statusCode	A	M	1	Specifies the result of the ‘SGSPServiceInfo’ this response corresponds to using the Status Code as specified in [BCAST12-Services].	unsignedByte

baseCID	A	M	0..1	The value to be filled in by the BSM as the value of the 'baseCID' attribute of the corresponding 'Service' or 'Content' fragment. If this attribute is omitted in the response, the BSA SHALL not instantiate the attribute 'baseCID' in the corresponding 'Service' or 'Content' fragment.	string
ProtectionKeyID	E1	O	0..N	The value to be filled in by the BSM as the value of the 'ProtectionKeyID' element of the corresponding 'Service' fragment. If this element is omitted in the response, the BSM SHALL not instantiate the element 'ProtectionKeyID' in the corresponding 'Service' fragment. This attribute SHALL not be instantiated when the corresponding request addresses 'Content' fragment using the 'GlobalContentID' element. Contains the following attribute: type	base64Binary
type	A	M	1	Type of ProtectionKeyID: For the possible values of this attribute, see 'Service' fragment specified in section 5.1.2.1.	unsignedByte
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

In order to be able to fill in information about possible service and content protection in 'Access' fragments, the SG-G of BSD/A needs to be able to request the corresponding parameters from the SGSS of BSM representing in turn access to the 'SP Management' of BSM. This request from SG-G to SGSS over SG-4 SHALL be an HTTP POST having an instantiation of the following 'SGSPAccessInfo' element as the payload.

Name	Type	Category	Cardinality	Description	Data Type
SGSPAccessInfo	E			Specifies the message for SG-G to request the values of the 'KeyManagementSystem' and 'EncryptionType' elements to be used in an 'Access' fragment from the SGSS. Contains the following attribute: globalServiceID Contains the following element: PrivateExt	
globalServiceID	A	M	1	Globally unique identifier of the service providing indirect identification of the access(es) in question. This is the value of the element 'GlobalServiceID' in the 'Service' fragment into the which 'Access' fragment in question is associated to.	anyURI
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	

<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	
------------------------	----	---	------	---	--

The following is the response message for the 'SGSPAccessInfo' to be used over the interface SG-4. The response from SGSS of BSM to SG-G of BSD/A SHALL be an instantiation of the following 'SGSPAccessInfoRes' message placed into the payload of the corresponding HTTP response.

Name	Type	Category	Cardinality	Description	Data Type
SGSPAccessInfoRes	E	M		Specifies the response message for a 'SGSPAccessInfo'. Contains the following attribute: statusCode Contains the following elements: KeyManagementSystem EncryptionType PrivateExt	
statusCode	A	M	1	Specifies the result of the 'SGSPAccessInfo' this response corresponds to using the Status Code as specified in [BCAST12-Services].	unsignedByte
KeyManagementSystem	E1	M	0..1	The value to be filled in by the BSD/A as the value of the 'KeyManagementSystem' element of the corresponding 'Access' fragment. If this element is omitted in the response, the BSA SHALL not instantiate the element 'KeyManagementSystem' in the corresponding 'Access' fragment. The element 'KeyManagementSystem' is specified in section 5.1.2.4.	complexType
EncryptionType	E1	M	0..N	The value to be filled in by the BSA as the value of the 'EncryptionType' element of the corresponding 'Access' fragment. If this element is omitted in the response, the BSA SHALL not instantiate the element 'EncryptionType' in the corresponding 'Access' fragment.	unsignedByte
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

#### 5.6.2.4 Service Provisioning Information Request

In order for SG-G of BSD/A to be able to provide provisioning fragments over interfaces SG-5 and SG-6, the SG-G needs to be able to request these from the SGSS of BSM. This request from SG-G to SGSS over SG-4 SHALL be an HTTP POST having an instantiation of the following 'SGProvisioningInfo' element as the payload.

Name	Type	Category	Cardinality	Description	Data Type
SGProvisioningInfo	E			Specifies the message for SG-G to request provisioning fragments from the SGSS. Contains the following elements: GlobalServiceID	

				GlobalContentID PrivateExt If both elements are omitted, the SGSS SHALL interpret this message requesting for 'PurchaseChannel' fragments.	
GlobalServiceID	E1	M	0..1	Globally unique identifier of the service for which the SGAS request provisioning fragments.	anyURI
GlobalContentID	E1	M	0..1	Globally unique identifier of the piece of content for which the SGAS request provisioning fragments.	anyURI
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

The following is the response message for the 'SGProvisioningInfo' to be used over the interface SG-4. The response from SGSS of BSM to SG-G of BSD/A SHALL be an instantiation of the following 'SGProvisioningInfoRes' message placed into the payload of the corresponding HTTP response.

Name	Type	Category	Cardinality	Description	Data Type
SGProvisioningInfoRes	E	M		Specifies the response message for a 'SGProvisioningInfoRes'. Contains the following attribute: statusCode Contains the following elements: ProvisioningFragment PrivateExt	
statusCode	A	M	1	Specifies the result of the 'SGProvisioningInfo' this response corresponds to using the Status Code as specified in [BCAST12-Services].	unsignedByte
ProvisioningFragment	E1	M	0..N	This element contains the SG data in question as a standard BCAST provisioning fragment.	complexType
PrivateExt	E1	O	0..1	An element serving as a container for proprietary or application-specific extensions.	
<proprietary elements>	E2	O	0..N	Proprietary or application-specific elements that are not defined in this specification. These elements may further contain sub-elements or attributes.	

## 5.7 Use of PreviewData to Support Service Switching, Browsing and Preview

Within the Service Guide, any or all of the Service, Content, Access, Schedule and 'PurchaseData' fragments MAY be associated with 'PreviewData' fragments. The 'PreviewData' fragment can include simple text, binary images, references to images, video clips and audio clips, or the combination of the aforementioned, or reference another service which could be a low bit rate version for the main service. The references to these media components of the preview data can be provided with remote URIs (for example, HTTP URL, RTSP URL). For the case of picture media component, the network MAY deliver the picture as a separate file over broadcast or over unicast, or directly within the PreviewData fragment itself by instantiating the PictureData element.

In case the media component is available via file delivery over the Broadcast or Interaction Channel, the Terminal MAY pre-fetch this component in order to accommodate the delays required by the network to deliver the preview data to the SG



application. How the Terminal decides on the media components to pre-fetch is out of scope of this specification and is application specific.

The following uses are specified for ‘PreviewData’ fragment:

- (1) Service-by-Service Switching: rendering the preview data associated with ‘PreviewData’ fragment
  - during the waiting time while terminal discovers of entry point to, and actually joins a service or content item, or
  - during the delay associated to user-initiated switching from one service/content item to another.
- (2) Service Guide Browsing: using the preview data associated with ‘PreviewData’ fragment as the representative item (eg. icon, logo) of a service, content, or purchase package.
- (3) Service Preview: presenting the preview data associated with ‘PreviewData’ fragment to users when the users want to briefly preview the service, content or purchase package;
- (4) Barker: presenting the preview data associated with ‘PreviewData’ fragment to users in place of a selected service or content to which the user does not have rights to access or is not subscribed.
- (5) Terms of Use: presenting the preview data associated with ‘PreviewData’ fragment as a representation of Terms of Use. This use of ‘PreviewData’ is allowed in a Service Guide fragment only when ‘TermsOfUse’ element is present in that fragment.
- (6) Alternative to blackout: presenting the preview data associated with ‘PreviewData’ fragment to users when the terminal has determined that the requested service is under blackout, as specified in [BCAST12-SvcCntProtection] section 7.1.

When a reference to ‘PreviewData’ fragment is made from another fragment for the uses (1), (2), (3), (4) or (6) above, the following applies:

- The referring fragment SHALL assign E1 element ‘PreviewDataReference’ with the fragment id of the referred ‘PreviewDataFragment’.
- The attribute ‘usage’ of the ‘PreviewDataReference’ in the referring fragment SHALL be assigned with value ‘1’, ‘2’, ‘3’, ‘4’ or ‘5’.

When a reference to ‘PreviewData’ fragment is made from another fragment for the use (5) above, the following applies:

- The referring fragment SHALL assign E2 element ‘PreviewDataIDRef’ with the fragment id of the referred ‘PreviewDataFragment’.
- The ‘PreviewDataIDRef’ in the referring fragment SHALL be a sub-element of ‘TermsOfUse’ E1 element.

It is possible that there are more than one ‘PreviewData’ fragments associated with the same Service, Content, Access, Schedule or PurchaseData fragment. In this case multiple instances of the ‘PreviewDataReference’ element will be present in the associated fragment; however the values of the ‘usage’ attributes of these ‘PreviewDataReference’ instances SHALL be mutually exclusive.

The following Table 7 summarizes the allowed uses of ‘PreviewData’ fragment when referenced by other Service Guide fragments for various usages. In this table, “X” means: the fragment that appears on the respective row can reference ‘PreviewData’ fragment for the usage stated on the respective column. The terminal SHALL support each of the identified uses of ‘PreviewData’ fragment, but MAY NOT support those unidentified uses of ‘PreviewData’ fragment. For instance, the Service fragment can reference the ‘PreviewData’ fragment as the representative item of the associated service for “Service Guide Browsing”, however while the ‘PreviewData’ fragment is associated with Service fragment with use “Service-by-Service Switching”, there is no guarantee that a BCAST 1.1 terminal supports such use.

	Service-by-Service Switching	Service Guide Browsing	Service Preview	Barker	Terms of Use	Blackout
Service fragment		X (as representative)	X (as preview of the service)	X (when no rights to view the	X (present only within element)	

		item of the service)		selected service)	'TermsOfUse')	
<b>Content fragment</b>		X (as representative item of the content)	X (as preview of the content)	X (when no rights to view the selected content)	X (present only within element 'TermsOfUse')	
<b>Schedule fragment</b>		X (as representative item of the schedule)	X (as preview of the schedule)		X (present only within element 'TermsOfUse')	
<b>Access fragment</b>	X					X
<b>PurchaseData fragment</b>		X (as representative item of the purchase data)	X (as preview of the purchase data)		X (present only within element 'TermsOfUse')	
<b>PurchaseChannel fragment</b>					X (present only within element 'TermsOfUse')	
<b>InteractivityData fragment</b>					X (present only within element 'TermsOfUse')	

**Table 7: Uses of 'PreviewData' fragment when referenced by other Service Guide fragments**

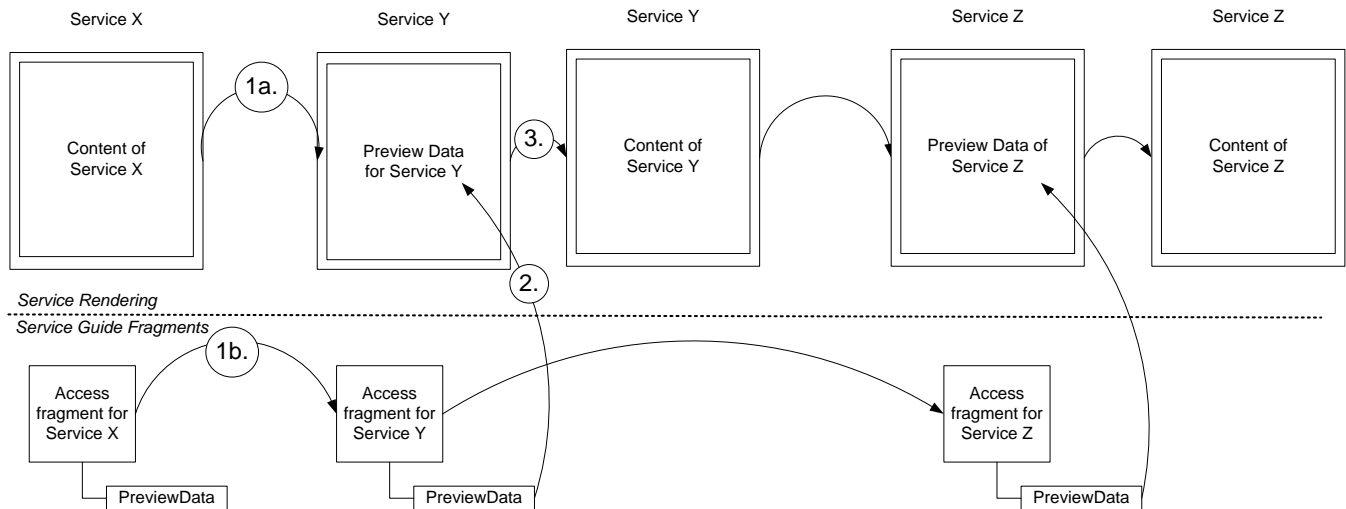
The details of the above usages of the preview data are further explained in the following sub-sections.

### 5.7.1 Preview Data for Service-by-Service Switching

When switching services, the terminal can render the preview data declared by 'PreviewData' fragment during the delay associated with terminal discovering of the entry point and actually joining to receive a service or content, or between user-initiated switching from one service or content to another. Which part or what media components of the preview data are used in rendering and with what layout is specific to each implementation. For example, a lower-end terminal could render just the descriptive text while a higher-end terminal could render the associated video clip.

This feature can be enabled by declaring the identifier of the 'PreviewData' fragment in the associated 'Access' fragment, and indicating the usage value as "1" (Service-by-Service Switching). Upon accessing content or a service, which has a related Access fragment with a reference to a 'PreviewData' fragment, the media components declared in this 'PreviewData' fragment SHOULD be received and rendered by the terminal, if available, before and until the main service appears.

The informative diagram Figure 8 below illustrates an example case how preview data supports Service-by-Service switching. In the diagram, the user first switches from Service X to Service Y (step 1a) and the terminal correspondingly switches from the Access related with Service X to the Access related with Service Y (step 1b). Consequently, the preview data referenced from Access Fragment related with Service Y is rendered (step 2). Finally, after stream initialization delay, content of Service Y is rendered (Step 3). At this point of time the preview data is removed from the screen. The following rendering for Service-by-Service switching is similar.



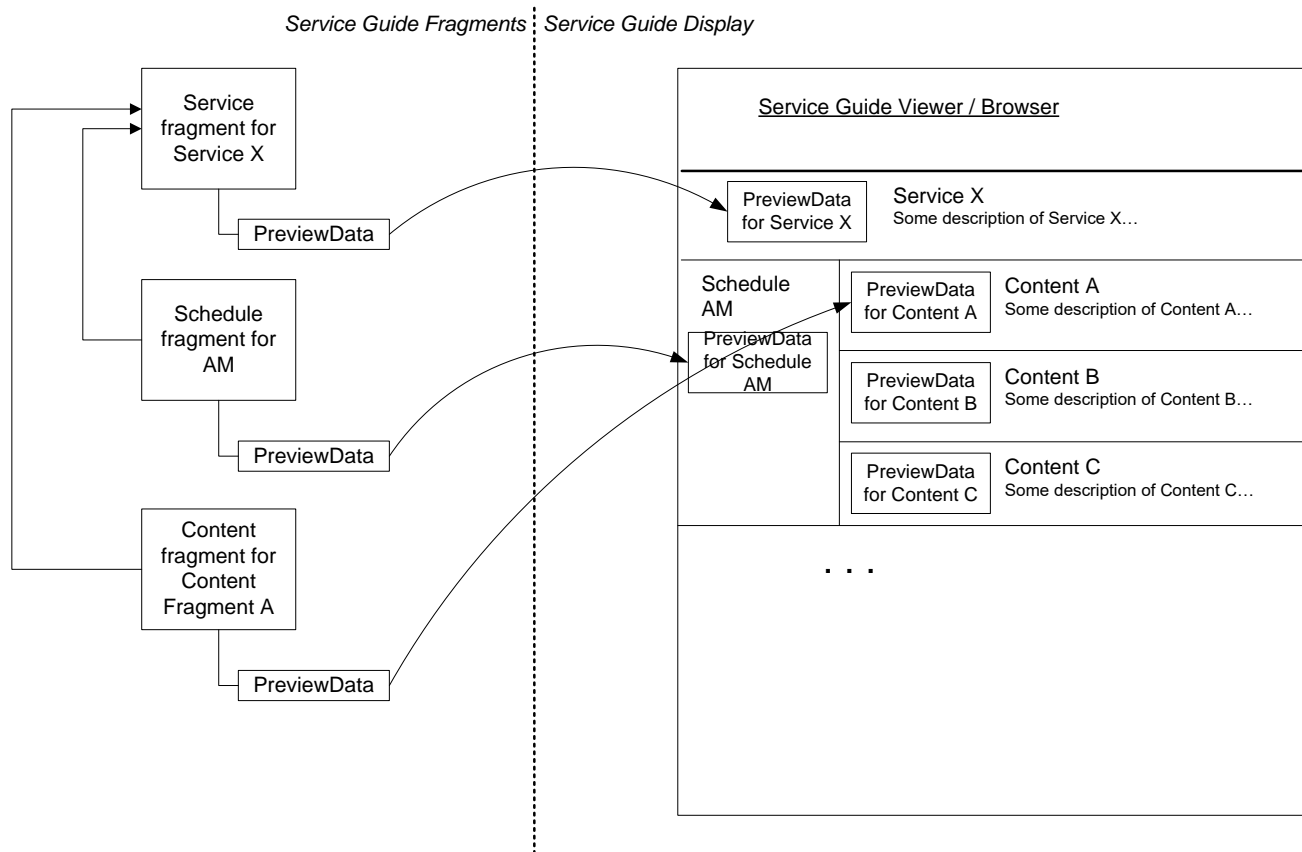
**Figure 8: Example case how preview data supports service-by-service switching**

### 5.7.2 Preview Data for Service Guide Browsing

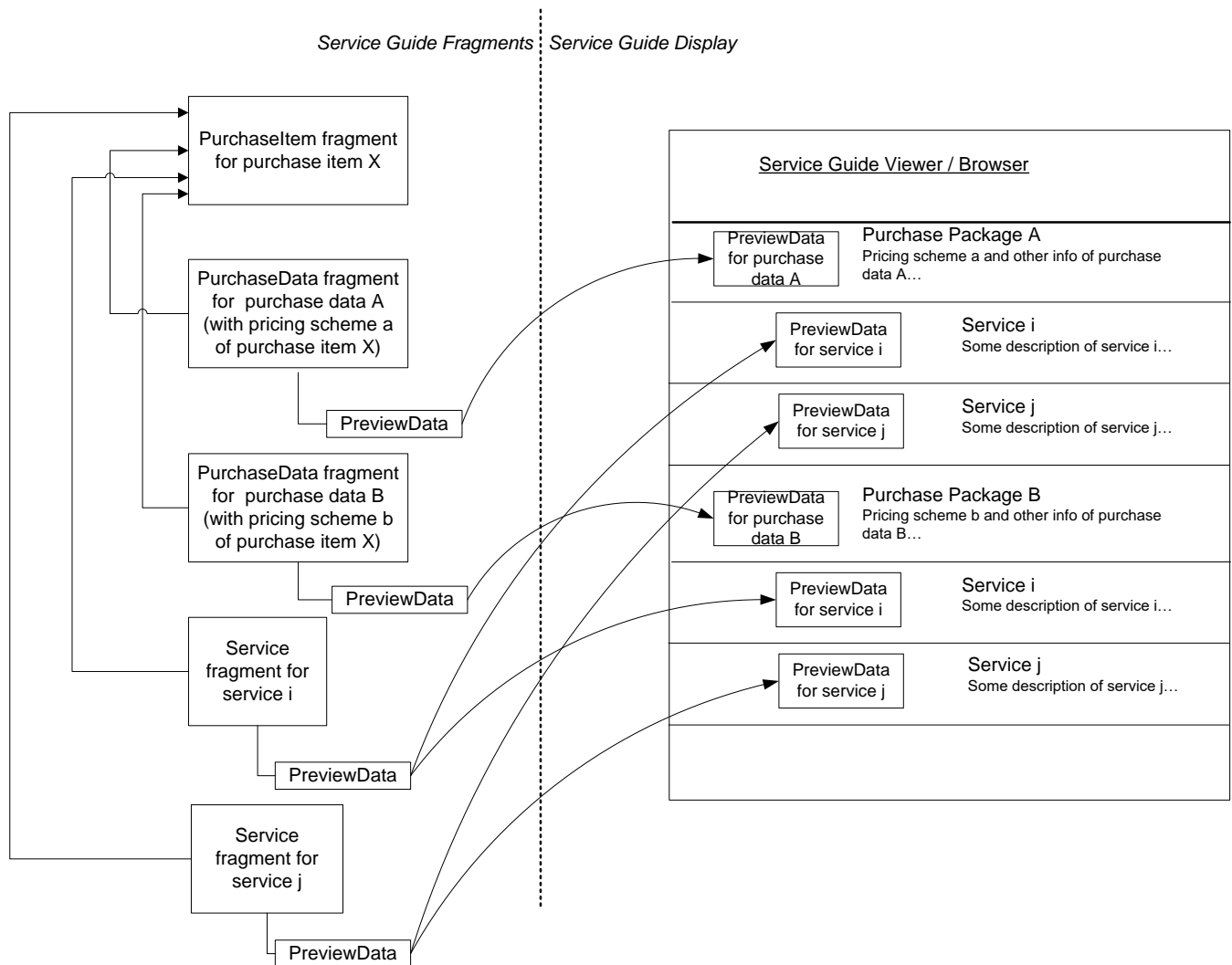
When browsing services, contents or purchase packages as a list, a tree or similar structure on the terminal, multiple services, contents, services or contents within a specific time slot, or purchase packages will be visible at the same time. In such case the terminal can render the associated preview data as the representative item (eg. icon, logo) of each service, content, schedule or purchase package. Which part or what media components of the preview data are used in rendering and with what layout is specific to each implementation. For example, a lower-end terminal could render just a small icon next to the service or content title while a higher-end terminal could render a simplified version of the actual service or content in a small size next to the service or content title and description.

This feature can be enabled by declaring the identifier of the ‘PreviewData’ fragment in the associated Service, Content, Schedule or PurchaseData fragments, and indicating the usage value as “2” (Service Guide Browsing).

The informative diagrams Figure 9 and Figure 10 below illustrate the support for Service Guide browsing with schedule view and purchase view. In Figure 9, the Service Guide browser displays multiple services and content in tree structure according to the scheduling. In this example the preview data of Service X, Content A, Content B, Content C and Schedule AM are shown as small icons next to the tile of the associated service, content and schedule. Similarly, in Figure 10 the Service Guide browser displays multiple services according to the purchase packaging. In this example, the two purchase packages corresponding to purchase data A and B are associated with the same purchase item X with different pricing scheme a and b respectively, hence the services belonging to purchase data A and B are identical (i.e. service i and service j), but the pricing scheme and other information of these two purchase data are different.



**Figure 9: Example case how preview data supports Service Guide browsing with schedule view**



**Figure 10: Example case how preview data supports Service Guide browsing with purchase view**

### 5.7.3 Preview Data for Service Preview

When viewing the detailed information of one specific service, content, schedule or purchase package on the terminal, the terminal can render the associated preview data to the users as an advance showing part (eg. trailer, posters) of the associated services, content, schedule or purchase package. Which part or what media components of the preview data are used in rendering and with what layout is specific to each implementation. For example, a lower-end terminal could render just a big poster image(s) while a higher-end terminal could render an A/V trailer in a big size or even full screen as the service preview.

This feature can be enabled by declaring the identifier of the ‘PreviewData’ fragment in the associated Service, Content, Schedule or PurchaseData fragments, and indicating the usage value as “3” (Service Preview).

The informative diagram Figure 11 below illustrates the support for service preview.

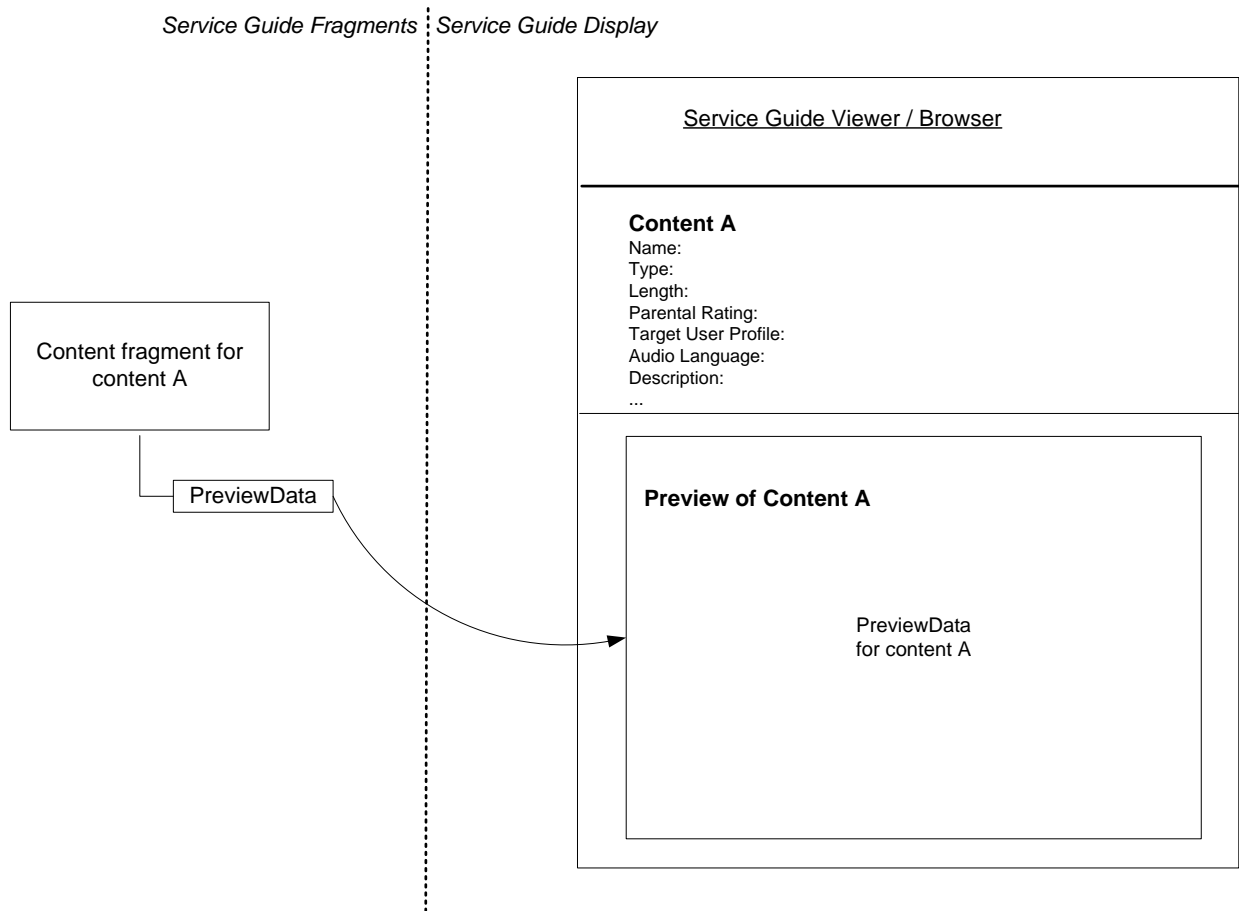


Figure 11: Example case how preview data supports service preview

### 5.7.4 Preview Data for Barker

The BCAST services can be provided with or without Service or Content Protection. Even when the service is protected and a particular user does not have rights to access the service, the preview data of the service can be visible in the Service Guide. When browsing the services or contents within the Service Guide the user might want to select the service or content which the user does not have rights to access or has not subscribed to, the preview data can be rendered in place of the actual service or content to give the user a preview on the selected service or content, or inform the user he/she does not has the rights to access or has not subscribed to the selected service or content.

This feature can be enabled by declaring the identifier of the ‘PreviewData’ fragment in the associated Service or Content fragments and indicating the usage value as “4” (Barker).

The informative diagram Figure 12 below illustrates the support for barker.

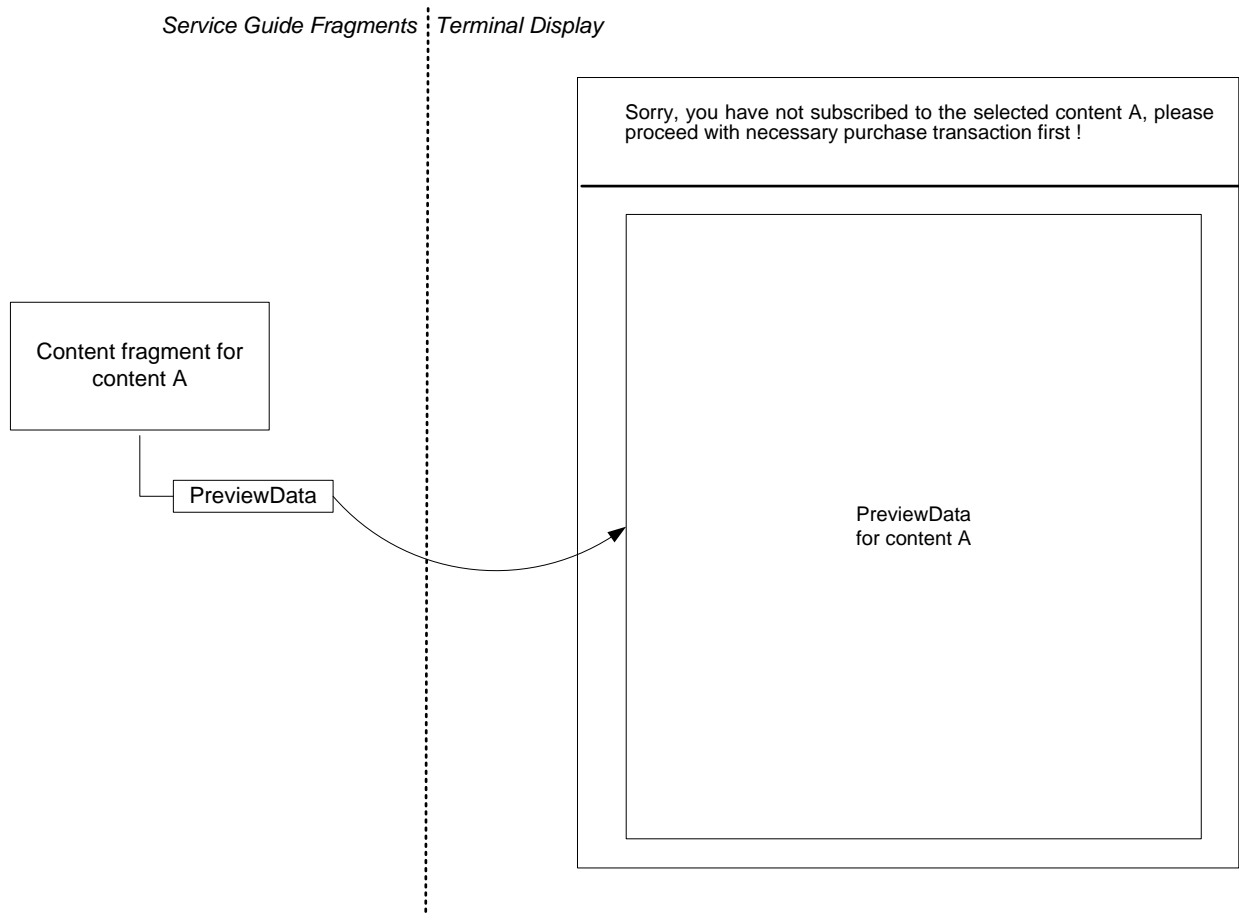


Figure 12: Example case how preview data supports barker

### 5.7.5 Preview Data as an alternative to blackout

Upon selection of a BCAST service by the end-user, the terminal fetches the corresponding Access fragment and performs the necessary steps to render the service. When the service is protected, the terminal may receive an STKM carrying an Access Criteria Descriptor enforcing a blackout in the current area of reception. In that case, the terminal has the possibility to check the Access fragment for a reference to a 'PreviewData' fragment with a 'usage' value of 5. If such 'PreviewData' fragment exists, it contains alternative payload the terminal can render instead of the blacked-out service. The 'PreviewData' fragment could also point to an Access fragment. In such a case the terminal can use that Access fragment to tune to an alternate service. The figure below illustrates these possibilities.

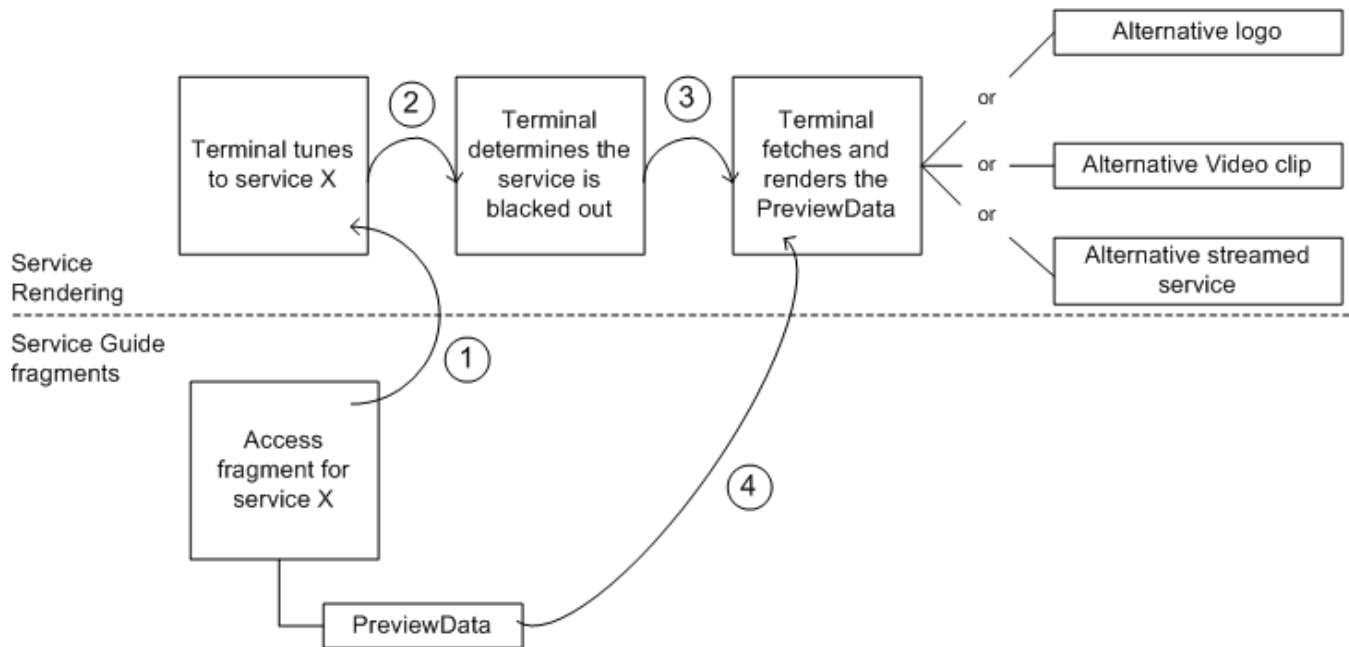


Figure 13: Example case how preview data supports alternative to blackout

## 5.8 Associating accesses with services and content

There can be different choices for the end users to consume services. These choices are represented using the ‘Access’ fragments associated to ‘Service’ fragments and (indirectly via ‘Schedule’ fragments) ‘Content’ fragments, the latter two representing services and individual pieces of the services respectively. The mechanisms for associating ‘Service’ and ‘Content’ fragments with ‘Access’ fragments, however, have different characteristics. Whereas a service is not always bound by time, a piece of content is. This is reflected by content having more specialized association to the access details than the service. The following sections define how the network instantiates Service Guide fragments in practice for signaling access parameters and how the terminal is expected to interpret them. Note that the distribution and presentation semantics for “Cachecast” services are different, and described separately in section 5.8.7.

### 5.8.1 Multiple accesses

Each ‘Access’ fragment represents different aspect of consumption of a service, or a content item belonging to that service, that ‘Access’ fragment is associated with. The network can give the terminal and to the end user different options for consuming services. In such a case the corresponding ‘Service’ or ‘Content’ fragment could be associated with more than one ‘Access’ fragment. The different options could be meant for the terminal to make the choice on behalf of the end user or by the end user himself/herself. In either case there needs to be a common understanding between the network and the terminal on how the choice is meant to be taken.

#### 5.8.1.1 Choices of access

When the network provides different means of accessing a service or content items of that service the network also needs to provide a meaning for each choice. Depending on the criterion for which each choice is based, the terminals first decides whether the selection is meant for terminal or the end user to make. In case the network provides multiple ‘Access’ fragments associated with ‘Service’ or ‘Content’ fragment, regarding those information contained in the ‘Access’ fragment meant for the terminal to use, transparent to the end user:

- the network SHALL ensure that these ‘Access’ fragments differ in value with respect to at least one of the following elements:
  - ‘AccessType’(e.g. ‘BDSType’)
  - ‘KeyManagementSystem’



- ‘EncryptionType’
- ‘TerminalCapabilityRequirement’
- ‘BandwidthRequirement’
- ‘ServiceClass’
- the terminal SHALL choose an ‘Access’ fragment that matches the best the terminal capabilities and/or settings (e.g. user settings). How the terminal actually makes the choice is out of scope of this specification.

The network SHALL declare the time interval when an ‘Access’ fragment applies as follows:

- Using the “t=” field of the SDP formatted Session Description instantiated in either the ‘Access’ or ‘SessionDescription’ fragment. In case the field has the value “t=0 0” it is considered to be an unbounded live stream.
- Using the ‘startTime’ and ‘endTime’ attributes of the ‘PresentationWindow’ element in a ‘Schedule’ fragment that the ‘Access’ fragment is associated with.
- Should the PresentationWindow be present, it is RECOMMENDED that the time period defined by <start time> and <stop time> in the Session Description encompasses the one defined by the PresentationWindow.

The terminal SHALL consider the latter first and in case no such ‘Schedule’ fragment or ‘PresentationWindow’ is available, the terminal SHALL consider the former.

## 5.8.2 Association between ‘Service’ and ‘Access’ fragment

The network can signal a service to be consumed by direct reference of one or more ‘Access’ fragment(s) to the ‘Service’ fragment representing the service. In such a case

- the terminal SHALL assume that the consumption method represented by the ‘Access’ fragment is available
  - during the interval constraint by the associated Session Description or
  - any time in case of absence of the constraint.
- In order for the terminal make a distinction between the different access methods the network SHALL take into account the constraints in the section 5.8.1 Multiple accesses and
- furthermore the network SHALL instantiate in the maximum one ‘Access’ fragment referring to the ‘Service’ fragment where the ‘Access’ fragment declares a notification component associated with the service using the element ‘NotificationReception’. The network MAY declare several ‘Access’ fragments referring to the ‘Service’ fragment given that not more than one ‘Access’ fragment contains such a notification component. Note: There is no meaningful way for the terminal to choose between two notification services and hence the restriction.

## 5.8.3 Association between ‘Service’ and ‘Access’ fragment through ‘Schedule’ fragment

In addition to the case described in the previous section, the network can also associate a ‘Service’ fragment with ‘Access’ fragments using ‘Schedule’ fragments. This mechanism is used to enable the concept of multi-language support as discussed in details in section 7. However, the basic intention is to represent a single language-specific choice of consumption to the end user. Therefore, the network SHALL instantiate each of the ‘Schedule’ fragments in question

- with the ‘ServiceReference’ element referring to the ‘Service’ fragment but
- without any instantiations of the element ‘ContentReference’
- Without any ‘onDemand’ attribute set to “true”.

In order for the terminal to make a distinction between the different access methods the network SHALL take into account the constraints in section 5.8.1 (Multiple accesses) and furthermore the network SHALL not associate any notification component with the service using the element ‘NotificationReception’ in the ‘Access’ fragment in question. If there are

multiple ‘Schedule’ fragments referring to the ‘Service’ fragment, at most one fragment SHALL have the ‘defaultSchedule’ attribute set to “true”.

## 5.8.4 Association between ‘Content’ and ‘Access’ fragment through ‘Service’ or ‘Schedule’ fragment

The ‘Content’ fragment provides means for the network to declare for the end user that a particular service is constructed of individual content parts. For example, traditional broadcast stations often make themselves known to the end users as a service but provide their services as individual pieces of content often also called as programs. The consumption of a single piece of content can be made available to the end user using the following four mechanisms:

- using a scheduled transmission over a broadcast access,
- using a scheduled transmission over an interactive access,
- using an end user initiated consumption of the content over the interaction channel (i.e. ‘OnDemand’) or
- using a (previously) downloaded file.

In the first two cases the availability of a single piece of content for the end user to consume is restricted implicitly. The stream for a particular content is available only during a specific interval of time and the network can also control when the terminals are allowed to access content over the interaction channel. The latter case, however, requires the terminal to control when the end user is allowed to consume the downloaded file.

### 5.8.4.1 Content access through service access

‘Content’ fragments can ‘inherit’ the ‘Access’ fragments that are associated with the ‘Service’ fragment. This is the use case where the network uses the same broadcast access for the service itself as for the individual pieces of content of the service. For this the network associates

- An ‘Access’ fragment directly with a ‘Service’ fragment as specified above in section 5.8.2 or
- An ‘Access’ fragment with a ‘Schedule’ fragment that is associated with the ‘Service’ fragment as specified above in section 5.8.3

In order to enable the terminal to interpret the use case correctly, each of the ‘Content’ fragments in question SHALL contain one and only one instantiation of the ‘ServiceReference’ element referring to the ‘Service’. In order for the terminal to represent these different methods of consumption for the end user to choose from, the terminal needs to comply with the following:

In the first case the terminal SHALL consider an ‘Access’ fragment (that is associated with a service) to be associated with a ‘Content’ fragment if

- the ‘Content’ and ‘Access’ fragment(s) in question refer to the same ‘Service’ fragment. In this case all those text languages or audio languages of content items in the ‘Content’ fragments (see section 7 Multi-language Support) which do not have their own dedicated ‘Schedule’ fragment (as outlines below or in section 5.8.4.2) are assumed to be accessible using the ‘Access’ fragment that is associated with the ‘Service’ fragment (either directly or through a ‘Schedule’ fragment as defined in section 5.8.2 or 5.8.3).
- and there is no ‘Schedule’ fragment associated with the same Content fragment that has the ‘onDemand’ attribute set to ‘true’

In the second case the terminal SHALL consider a ‘Schedule’ fragment (that is associated with a service only) and its associated ‘Access’ fragment to be associated with a ‘Content’ fragment

- If the ‘Content’ and ‘Schedule’ fragment(s) in question refer to the same ‘Service’ fragment, the ‘Content’ fragment refers to one and only one ‘Service’ fragment
- There is no ‘Schedule’ fragment associated with the same Content fragment that has the ‘onDemand’ attribute set to ‘true’.

### 5.8.4.2 Content specific access via ‘Schedule’ fragment

There can be cases where it is desired for an individual piece of content of a service to have access methods specific only to the individual piece of content. These access methods are associated with the content using ‘Schedule’ fragments referring to the ‘Content’ and ‘Access’ fragments referring to the ‘Schedule’ fragments in turn. Similarly as in the case of service access described above in section 5.8.2, the network needs to comply to the following rules in order for the terminals to interpret this use case correctly. In such a case the network SHALL instantiate a ‘Schedule’ fragment in accordance with the following:

- with the ‘PresentationWindow’ element declaring the available time period of rendering. This PresentationWindow element SHOULD have the startTime and endTime attribute matching the “t=” field in the SDP formatted Session Description information instantiated by either the ‘Access’ fragment (referencing this ‘Schedule’ fragment) or the ‘SessionDescription’ fragment, with the exception when the time field is set to “t=0 0”.
- with the ‘ContentReference’ element referring to the ‘Content’ fragment but
- without instantiation of the element ‘InteractivityReference’.
- and the ‘Content’ fragment and the ‘Schedule’ fragment SHALL refer to the same ‘Service’ fragment.
- in order for the terminal to make a distinction between the different access methods the network SHALL take into account the constraints in the section 5.8.1 Multiple accesses and
- furthermore the network SHALL not associate any notification component with the service using the element ‘NotificationReception’ in the ‘Access’ fragment in question.
- in addition, the ‘contentLocation’ attribute of the ‘ContentReference’ element MAY be instantiated even if both broadcast channel ‘Access’ fragments and interaction channel ‘Access’ fragments are referencing the same ‘Schedule’ fragment. Although, for streaming delivery, the ‘contentLocation’ attribute is then only used for the interaction channel access, and not for the broadcast channel access.

The terminal in turn SHALL expect that the consumption method(s) represented by the ‘Access’ fragment(s) can only be used during the time period restricted by the ‘PresentationWindow’ element of the ‘Schedule’ fragment.

Furthermore the network MAY set the ‘onDemand’ attribute to “true” in the ‘Schedule’ fragment.

- In case there are one or more ‘Schedule’ fragments related to a ‘Content’ fragment that have the ‘onDemand’ attribute set to ‘true’, the terminal SHALL expect that that content item is only accessible through those ‘Schedule’ fragments, i.e. the ‘Access’ fragment associated with the ‘Service’ fragment, which is associated with this ‘Schedule’ fragment and related ‘Content’ fragment is not valid for that ‘Content’ item.
- and in case the ‘onDemand’ attribute is false or not declared in any of the ‘Schedule’ fragments referring to the ‘Content’ fragment, the terminal SHALL expect that the content item is also accessible through the ‘Access’ fragment associated with the ‘Service’ fragment as described in section 5.8.4.1.

#### 5.8.4.2.1 PresentationWindow, StartTime and EndTime

When a ‘Content’ fragment is associated with a ‘Schedule’ fragment the terminal is provided a pair of time intervals

- one interval declared by the ‘StartTime’ and ‘EndTime’ elements of the ‘Content’ fragment and
- another declared by the ‘PresentationWindow’ of the ‘Schedule’ fragment.

The former is intended for direct display to the end user the fixed interval during which the corresponding piece of content can be consumed by the end user. The latter may either be used to signal to the terminal when the specific access method represented by the ‘Schedule’ fragment and the associated ‘Access’ fragment is to be used.

In case the interval declared by the ‘PresentationWindow’ does not cover the entire interval declared by the ‘StartTime’ and ‘EndTime’ elements of the ‘Content’ fragment, the terminal SHALL cover the remaining time using one of the ‘Access’ fragments associated with the ‘Service’ fragment the ‘Content’ fragment is associated with or using one of the ‘Access’ fragments associated with other ‘Content’ fragments of the same service. In case no such ‘Access’ fragment is available there are no expectations defined by this specification for the terminal. If in turn there are multiple such accesses available the terminal SHALL choose as specified in sections 5.8.1.1.

## 5.8.5 Use of ‘defaultSchedule’ and ‘onDemand’ attributes

The usage of the ‘defaultSchedule’ and ‘onDemand’ attribute defines the access method that the terminal has to take into account when entering the ‘Service’ and the access method that takes precedence.

The ‘defaultSchedule’ attribute indicates the access method that takes precedence in case multiple access methods are defined for a certain content item. If there are multiple ‘Schedule’ fragments referring to the ‘Content’ fragment, at most one fragment MAY have the ‘defaultSchedule’ attribute set to ‘true’. A ‘Schedule’ fragment that has the ‘defaultSchedule’ attribute set to ‘true’ and has a link to the ‘Content’ fragment takes precedence over the ‘Schedule’ fragment that has the ‘defaultSchedule’ attribute set to ‘true’ but only has a link to the ‘Service’ fragment of which that ‘Content’ fragment is associated to. It should be noted that specific terminal (language) settings may override the ‘defaultSchedule’.

The ‘onDemand’ attribute indicates the scheduled transmission of content items that are accessible over the unicast channel where the transmission is not necessarily bounded into any specific interval of time. This type of content typically represents the "on-demand" component of a service. If this attribute is present and “true” it indicates that the referenced content item SHALL NOT be accessed automatically through this ‘Schedule’ fragment by the terminal when the associated service is selected and this ‘Schedule’ fragment is valid. This attribute in the ‘Schedule’ fragment MAY be set to ‘true’ if and only if the ‘Access’ fragments associated to the ‘Schedule’ fragment do not have the ‘BroadcastServiceDelivery’ element instantiated. In section 5.8.4.1 it is described how a ‘Content’ fragment can ‘inherit’ the ‘Access’ fragments that are associated with the ‘Service’ fragment. However, in the case the ‘onDemand’ attribute of a ‘Schedule’ fragment associated with a content item is ‘true’ the terminal SHALL assume that the ‘Access’ fragment associated with the ‘Service’ fragment is NOT applicable for a certain ‘Content’ fragment. In the case a ‘Content’ fragment has multiple ‘Schedule’ fragments associated with it, but only one or a subset of those ‘Schedule’ fragments have the ‘onDemand’ attribute set to ‘true’ the terminal SHALL ignore all ‘Schedule’ fragments that have the ‘onDemand’ attribute set to ‘false’ (either declared or by absence of it).

With regard to the combination of ‘defaultSchedule’ and ‘onDemand’ the following applies:

- for ‘Content’ fragments that have one or more ‘Schedule’ fragments associated with it that have the ‘onDemand’ attribute set to ‘true’, the ‘Schedule’ fragment with the ‘defaultSchedule’ set to ‘true’ determines the favourable access method of that content item. The terminal MAY use this information to facilitate the end user selection between the options of the various applicable access methods.
- for ‘Content’ items that are only associated with ‘Schedule’ fragments that all have the ‘onDemand’ attribute set to false (either declared or by absence of it) the terminal SHALL interpret the access method represented by the ‘Schedule’ fragment to be applicable without possible end user intervention. In this case the ‘Schedule’ fragment associated with the ‘Content’ fragment that has the ‘defaultSchedule’ attribute set to ‘true’ takes precedence. If none of the ‘Schedule’ fragments referring to the ‘Content’ fragment have the ‘defaultSchedule’ attribute set to ‘true’ the terminal SHALL use the ‘Access’ fragment that is associated with the ‘Service’ fragment as the default (see section 5.8.4.1).

## 5.8.6 Terminal interpretation for content consumption

When the terminal receives the complete Service Guide, the terminal will find it has a collection of ‘Service’ fragments and for each ‘Service’ fragment there is a set of ‘Content’ fragments, ‘Schedule’ fragments and ‘Access’ fragments associated with it. It is important to note that the elements and attributes provided by the ‘Service’ and ‘Content’ fragments provide information how particular content can be offered to the end user for consumption. The ‘Schedule’ and ‘Access’ fragments on the other hand provide the true availability and access methods of content. From a terminal point of view, whenever a service is selected by the end user, the terminal will find itself with a set of ‘Schedule’ and/or ‘Access’ fragments. This set of ‘Schedule’ and/or ‘Access’ fragments can be considered to have a time order where at a certain point in time multiple fragments can be valid. This is regardless of whether these ‘Schedule’ fragments do or do not have a specific reference to a ‘Content’ fragment associated with the corresponding service. Therefore whenever a service is selected the terminal has to access the content through the ‘Access’ fragment that is applicable at that particular point in time. Recall section 5.8.1.1, Choices of access.

At any point in time the terminal might find there are more ‘Access’ fragments valid for the service because of one or more of the following use cases:

- Both unicast and broadcast accesses are available for the service,

- multiple text and/or audio languages are available for the service or
- multiple text and/or audio languages are available for a specific piece of content at that point in time.

The consumption of a single piece of content that is part of a service can be made available for the end user in the following ways:

- automatically without user intervention upon entering the service by terminal selection of the access method that is valid at the point in time, based on the content items that are part of the regular programming, as indicated by Schedule fragments that have the 'onDemand' attribute set to 'false' (either declared or by absence of it)
- upon user selection.

In the first case terminal SHALL only be required to make a selection from 'Access' fragments that are associated:

- with the 'Service' fragment as described in section 5.8.2.
- with the 'Service' fragment through a Schedule fragment as described in section 5.8.3.
- with 'Schedule' fragments related to 'Content' fragments that are not part of the "on demand" content, which are defined by having the 'onDemand' attribute set to 'false' (either declared by absence of it).

to determine which 'Access' fragment is valid and takes precedence and MAY be used for automatic content consumption.

In the latter case, access based on user selection is applicable for:

- 'Content' fragments that have the 'Schedule' fragment related to it with the 'onDemand' attribute set to 'true'
- 'Content' fragments that have 'Schedule' fragments related to it that have the both the 'onDemand' and the 'defaultSchedule' attributes set to false (either declared or by absence of it).

Whenever there are multiple text or audio languages available the terminal can give the end user the option to make a selection. The terminal can for example highlight or place blinking icons with the content in the Service Guide that has at that point in time an applicable access associated with it. Based on the specific terminal language settings or based on the 'Schedule' fragment that is declared to have the 'defaultSchedule' attribute set to 'true' the terminal can make a decision which 'Access' fragment provides the preferable access parameters for the service.

It is important to note that the 'StartTime' and 'EndTime' elements in the 'Content' fragment do not declare any validity of the content whatsoever, these elements are purely intended to present to the end user enabling the terminal to tell the end user which piece of content is currently being played. In a typical case the network could declare a set of 'Content' fragments associated with the same 'Service' fragment, with the 'StartTime' element and 'weight' attribute set to represent a chronological order of pieces of content belonging to a service. The 'StartTime' of a 'Content' fragment could then implicitly serve as the 'EndTime' of the 'Content' fragment with the next lower 'weight' attribute. For the cases above it is very well possible that a certain 'Access' fragment that is associated with a 'Content' fragment, is applicable during an interval that mismatches with the time window, as defined by the 'StartTime' and 'EndTime' elements (either explicit or implicit declared), of that 'Content' fragment. For instance it could have a partial overlap of a time window of other 'Content' fragments belonging to the same service.

Despite the Service Guide having coherent structure there can be three cases in which the terminal will encounter unexpected events:

- In the event that by accident the Service Guide does have overlapping 'Schedule' fragments with the 'onDemand' attribute set to 'false' (either declared by absence of it) and that are associated to different 'Content' fragments the terminal might not be able to determine the 'Access' fragment that is valid and should be selected, i.e. multiple 'Access' fragments are valid and the terminal can not uniquely decide which one takes precedence over the other based on the 'defaultSchedule' attribute. In such a case the terminal MAY assume that of these overlapping 'Schedule' fragments the 'Schedule' fragment which has the 'PresentationWindow' element with the earliest 'startTime' attribute to take precedence and accordingly the 'Access' fragment associated with this 'Schedule' fragment.
- In the event that a certain 'Access' fragment is selected but there is no IP stream available on the declared IP address in the 'SessionDescription' fragment. This can e.g. be the case when the live broadcast session schedule has changed, but the Service Guide could not be updated, or just because the access is currently unavailable e.g. due to radio

reception conditions. In such a case the terminal MAY select another applicable 'Access' fragment. In case no such 'Access' fragment is available the terminal MAY present an error to the end user. In any case the terminal SHOULD be able to handle such a case and not turn to an error-state.

- In the event that a 'Content' fragment has multiple text and audio languages declared, where all audio and text languages are associated with 'Schedule' fragments that directly reference that 'Content' fragment, but none of these 'Schedule' fragments have the 'defaultSchedule' attribute set to 'true' and none of these 'Schedule' fragments have the 'onDemand' attribute set to 'true' and none of the languages match the specific terminal language settings. In this case the terminal SHALL use the 'Access' fragment that is associated with the 'Service' fragment (which is referenced by that 'Content' fragment, see section 5.8.4.1).
- In the event a 'Content' fragment has multiple 'Schedule' fragments associated with it, but only one or a subset of those 'Schedule' fragment have the 'onDemand' attribute set to 'true'. In this case the terminal SHALL ignore all 'Schedule' fragment that have set the 'onDemand' attribute to 'false' (either declared or by absence of it).

### 5.8.7 Distribution Window and Presentation Window for Cachecast Services

In addition to streaming services, for which the rendering time is the same as the delivery time, the SG also supports file delivery services for which the network can distribute the content prior to rendering. For file distribution services, the terminal downloads the content beforehand and stores it for later user consumption. Two types of file delivery services are possible: those which pertain to scheduled rendering, such as Cachecast, and others which are not associated with scheduled rendering, such as music or video downloads. The temporal semantics of Cachecast services are specified by the Distribution Window and Presentation Window elements in the Schedule fragment. Note in particular that for Cachecast, neither window is necessarily fixed in time or size:

- The terminal can begin the download as early as 'startTime' of Distribution Window; the 'startTime' does not represent the default start time of the download.
- The terminal can begin the download as late as 'endTime' of Distribution Window and should still be able to acquire the entire file within the 'duration' of the Distribution Window; the 'endTime' does not represent the default end time of the download.
- The user can begin rendering as early as the 'startTime' of the Presentation Window; the 'startTime' does not represent the default start time of rendering;
- The user can begin rendering as late as the 'endTime' of the Presentation Window, and still be able to expect the rendering to run to completion; the 'endTime' does not represent the default end time of rendering. The presentation duration is given by the 'duration' of Presentation Window.

## 6. Discovery of Service Guide

### 6.1 Service Guide Transmitted over Broadcast Channel

The service guide discovery mechanisms that are specified in this section relate to the discovery of a Service Guide that is distributed over Broadcast Channel (i.e. they do not relate to Service Guides that are not using the Broadcast Channel). In this case, the Terminal will need to know how to find and access the broadcast IP flows that carry the broadcast Service Guide.

#### 6.1.1 Service Guide Discovery over Broadcast Channel

When the Service Guide is delivered using the broadcast channel the Service Guide Announcement Channel is thought as the starting point of the retrieval. Recall that the Service Guide Announcement Channel provides all the information the terminals need for retrieving the Service Guide. Therefore to discover the Service Guide the terminals basically need to locate the file delivery session carrying the Service Guide Announcement Channel. The access parameters of the FLUTE session representing the Service Guide Announcement Channel are called the entry point to a Service Guide on a Broadcast Channel.

In one broadcast area there MAY exist multiple Service Guides and any number of these MAY be delivered simultaneously using the broadcast channel. In such a case, in principle, it is the responsibility of the underlying BDS to provide the signalling of the entry points of the Service Guides to the terminals. However, if such a signalling is not available or being used, the following parameters SHALL be used as the entry point:

- (OPTIONALLY) IP Source Address
- Fixed Destination Multicast IPAddress: 224.0.23.165 for IPv4 or FFOX:0:0:0:0:0:132 for IPv6
- Fixed Destination Port: 4090

The terminal SHALL assume that

- there is at most one FLUTE session per entry point.
- the value of the Transport Session Identifier (TSI) MAY be any valid value and the number of ALC/LCT channels in the FLUTE session for Service Guide announcement is fixed to 1.

If the underlying BDS supports specific signalling of the Service Guide entry points the terminal SHALL expect the BDS also to provide the specific signalling. The detailed guidelines for such signalling in specific BCAST Distribution Systems are given in the BDS Adaptation Specifications (See [BCAST12-DVBH-IPDC-Adaptation], [BCAST12-MBMS-Adaptation] [BCAST12-BCMCS-Adaptation], [BCAST12-DVBSH-IPDC-Adaptation], [BCAST-12-FLO-Adaptation] and [BCAST12-NGH-Adaptation]).

The terminal SHALL support the initial Service Guide discovery over Broadcast Channel.

#### 6.1.2 Service Guide Discovery over Interaction Channel

The entry point to a Service Guide on an Interaction Channel SHALL be defined as the URL to a file containing Session Description or URL to a resource resolving to a Session Description which describes the file distribution session carrying Service Guide announcement information and possibly Service Guide. This file distribution session originates from Service Guide Generation Function and Service Guide Distribution Function (over interface SG-5). The entry point to a Service Guide on an Interaction Channel MAY be either fixed, or provisioned to the terminal (e.g. through BDS specific signalling), requested from interactive bootstrap URL (i.e., through the key “SGEntryPointsOnly” in section 5.4.3.3) or provided out-of-band (e.g. through a public or private web site).

Within a single BDS, there MAY be different Service Guide generated for different service coverage areas, requiring a different entry point for each particular service coverage area. It is not in scope of this specification to define how the device learns about the applicable URL.

The terminal with interaction channel SHALL support the initial Service Guide discovery over Interaction Channel.

## 6.2 Service Guide Transmitted over Interaction Channel

The service guide discovery mechanisms that are specified in this section relate to the discovery of a Service Guide that is to be distributed over Interaction Channel. The Terminal needs to get some discovery information, and sends the request to acquire Service Guide.

The entry point to Service Guide acquisition over Interaction Channel SHALL be a URL which indicates the location of Service Guide. Example of such URL is (<http://provider.com/serviceguide>). This is the address that the SG-C in the Terminal accesses in order to get Service Guide data over Interaction Channel as specified in section 5.4.3. There are several possible ways terminal can get the entry point information. The Terminal SHALL support the following three means:

- the entry point information is provided using the 'AlternativeAccessURL' element of SGDD; and;
- the entry point information is provisioned to the Terminal via Terminal Provisioning function.
- the entry point information is provided using the "UnicastServerURL" element of SGDD, within the "SGEntryPoint" parent;

The Terminal MAY support the following means:

- the entry point information is acquired by the Terminal via the DNS SRV mechanism.

For the first case, considering that it has no prior knowledge of the 'AlternativeAccessURL', the terminal performs a two-step discovery operation, by first discovering the Service Guide transmitted over the broadcast channel, and then requesting those fragments that are declared, in the retrieved SGDDs, as being available from the interaction channel. Note that, in this case, retrieval of Service Guide fragments over the interaction channel is scoped by the SGDDs transported over the broadcast channel, unless the terminal issues an unspecific Service Guide request, and the server is able to answer such request.

For the second case the terminal SHALL support OMA BCAST Management Object parameter '<X>/SGServerAddress/' as specified in [BCAST12-Services]. Further the entry point information MAY be fixed in the Terminal or provided out-of-band by the means not specified in this specification (such as WAP PUSH, SMS, MMS, Web page, user input, etc).

For the third case, the terminal performs a two-step discovery operation. The terminal first retrieves SGDD over the broadcast channel or interaction channel with SG entry point information. For retrieving the SGDD over interaction channel, the destined URL, named "interactive bootstrap URL", is fixed or provisioned by OMA BCAST MO or provided out-of-band by the means not specified in this specification. Note that "interactive bootstrap URL" and the entry point to Service Guide acquisition over Interaction Channel, SHOULD NOT be provisioned simultaneously in the same given terminal.

Then the terminal requests for SGDDs and/or SGDUs from the unicast server signalled in the "SGEntryPoint" element of SGDD. The terminal SHALL ignore 'AlternativeAccessURL' and SHALL use 'UnicastServerURL' instead, if there is a <SGentrypoints> element instantiated in the SGDD including at least one UnicastServerURL, and either scoped by no BSMselector or scoped by the BSMselector identifying the BSM which the terminal is affiliated to.

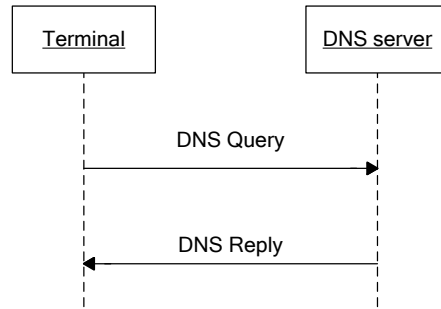
For the fourth case the Terminal utilizes the DNS SRV mechanism [RFC 2782] for entry point discovery as described in Section 6.2.1.

### 6.2.1 Service Guide Discovery using DNS

In this case, the domain name and port of the entry point URL is discovered using the DNS SRV mechanism [RFC 2782] with the following input parameters:

- Service: Defined as "oma-bcast-sg".
- Protocol: Set to value "tcp". Specifies the protocol to contain the particular service.
- Domain name: the domain for which the returned records are valid. The value is RECOMMENDED to be derived from MCC (Mobile Country Code) and MNC (Mobile Network Code) retrieved from the USIM, as described in section 6.2.1.1. If not possible, the domain name MAY be derived from the network attachment phase (DHCP server), or from a value fixed in the Terminal.





**Figure 14: Discovery of domain name part of the entry point URL with DNS**

The output of the DNS SRV lookup is an ordered list of target domain names and corresponding port numbers, each pair pointing to a Service Guide server available within the domain specified in the DNS query. The URL of the entry point to Service Guide acquisition is derived according to the following syntax in ABNF [RFC4234]:

```

sg-entry-url = "http://" target_domain "." target_port "/bcast-service-guide"
target_domain=<target field of the DNS SRV record>
target_port=<port field of the DNS SRV record>
  
```

For example, a DNS SRV record including *provider.com* as target and *8080* as port the Service Guide entry point URL resolves to the following:

```
http://provider.com:8080/bcast-service-guide
```

### 6.2.1.1 Derivation of Domain Name from IMSI

For 3GPP systems the UE MAY derive the domain name from the IMSI as described in the following steps:

1. Take the first 5 or 6 digits, depending on whether a 2 or 3 digit MNC is used (see [3GPP TS 31.102]) and separate them into MCC and MNC; if the MNC is 2 digits then a zero shall be added at the beginning.
2. Use the MCC and MNC derived in step 1 to create the "mnc<MNC>.mcc<MCC>.pub.3gppnetwork.org" domain name.
3. Add the label "bcast." to the beginning of the domain.

An example of a domain name to be used in a DNS SRV lookup is:

```
IMSI in use: 234150999999999;
```

where:

- MCC = 234;
- MNC = 15; and
- MSIN = 0999999999,

which gives the domain name: `bcast.mnc015.mcc234.3gppnetwork.org`.

## 7. Multi-language Support

This section defines how the language information conveyed in the Service Guide can be used both to specialize the presentation of the Service Guide itself for individual languages and to provide language specific options for consuming the services declared by the Service Guide. Whereas the former is in principle about textual descriptions provided in individual languages, the latter also considers possible language dependent audio tracks and sub-titles. The section is organized to address the specialization of the Service Guide regarding how to declare the language specific options of consuming the services. It should be noted that the technical details of enabling multi-language support are provided in conjunction with the corresponding attributes and elements of the Service Guide and that the following sections merely provide logically meaningful binding of these individual details.

### 7.1 Declaring service guide elements in different languages in the Service Guide

The Service Guide may provide both technical information not visible to the end user and purely descriptive information meaningful only to the end user. Whereas the former part is independent of such things as the nationalities of the end users, the latter can be explicitly specialized for end users with different nationalities. The data model of the Service Guide enables the network to provide textual descriptions in the Service Guide multiple times, one description per language. This is simply achieved by allowing an individual element representing a textual description to be instantiated multiple times and specializing each of the individual instances of these elements using the 'xml:lang' attribute defined by XML standard itself.

In case an individual element of the Service Guide represents textual description intended only for the UI purposes the following applies:

- In case the element is instantiated more than once in an individual SG fragment, the network **MUST** include the 'xml:lang' attribute to each of the instances of the element in the SG fragment where
  - each of the instantiated 'xml:lang' attributes **MUST** have a different value.
  - an instantiation of the 'xml:lang' attribute with the value of an empty string **SHALL** be interpreted by the Terminal as a default element applicable for language settings which are not targeted by any 'xml:lang' attribute in sibling elements of the same type.
- On the other hand if the 'xml:lang' attribute is omitted in the element the terminal **SHALL** interpret the element to be applicable independent of any language settings.

Table 8 below summarizes the elements that can be specialized for different languages.

	Service	Content	Purchase Item	Purchase Channel	Purchase Data	Preview Data	Interactivity Data	SGDD
Name	X	X	X	X				
Description	X	X	X	X	X			
Extension.Description	X	X	X	X	X		X	
AudioLanguage	X	X						
TextLanguage	X	X						
Genre	X	X						
BroadcastArea.TargetArea.name_area	X	X						
PromotionInfo.Title					X			
PromotionInfo.Description					X			
PromotionInfo.PromotionExtension.Description					X			
PromotionInfo.Title					X			
Video.AlternativeText						X		
Audio.AlternativeText						X		

Picture.AlternativeText						X		
Text						X		
InteractivityType							X	
BSMList.BSMSelector.Name								X
DescriptorEntry.GroupingCriteria. GenreGroupingCriteria								X

Table 8: Elements that can be specialized for different languages

## 7.2 Declaring components of the broadcast service in different languages in the Service Guide

The services, declared by ‘Service’ fragments, as well as individual parts of the services, declared by ‘Schedule’ and ‘Content’ fragments, are possible to be provided in multiple ways to be consumed. Language is one possible criterion for end users to differentiate and consume various services. In order for the end users to understand the meaning of the different options, the Service Guide needs to be able to represent the criterion as well as the individual choices for the end user. For this purpose the Service Guide provides specific elements and attributes that can be used to signal the semantics of the different choices, in this case the specific languages.

### 7.2.1 Service and content specialization with audio tracks and sub-titles

The network can declare both services as well as individual pieces of services, represented by a ‘Service’ and ‘Content’ fragments respectively, to be available in different languages

- with multiple audio streams and/or
- with multiple sub-title streams.

In the Service Guide the available language is declared by the ‘AudioLanguage’ and/or ‘TextLanguage’ elements of the ‘Content’ and ‘Service’ fragments respectively. The association to the specific details of consumption is in turn achieved through ‘languageSDPTag’ attributes of the ‘AudioLanguage’ and ‘TextLanguage’ elements. When instantiated, the values of these attributes correspond to values found in the media sections (m=audio..., m=video..., etc.) in SDP files. Where the SDP files are associated with ‘Content’ and ‘Service’ fragments via ‘Schedule’ and ‘Access’ fragments as specified in section 5.8, the exact association between a value of a ‘languageSDPTag’ and a media section is signalled as follows:

The terminal SHALL associate a particular ‘AudioLanguage’ element to a media section in SDP file if and only if

- the media section is of the type ‘m=audio...’ and
- the media section contains an attribute line “a=lang:<language tag>” where the value in place of ‘<language tag>’ is an exact match of the value of the ‘languageSDPTag’ attribute in the ‘AudioLangauage’ element in question.

The terminal SHALL associate a particular ‘TextLanguage’ element is associated to a media section in SDP file if and only if

- the media section is of the type ‘m=video...’ and
- the media section contains an attribute line ‘a=rtpmap’ with the media subtype ‘3gpp-tt’ and
- the media section contains an attribute line “a=lang:<language tag>” where the value in place of ‘<language tag>’ is an exact match of the value of the ‘languageSDPTag’ attribute in the ‘TextLangauage’ element in question.

In case a ‘Content’ or ‘Service’ fragment is associated with a ‘Schedule’ fragment having the ‘defaultSchedule’ attribute instantiated and the ‘Schedule’ fragment is associated to an SDP file having multiple media sections for either audio or subtitle tracks, then the choice between the media sections to be applied is out of scope of this specification.

Recall that a ‘Content’ fragment can be associated with a ‘Schedule’ fragment either

- directly by the ‘Schedule’ fragment referring the ‘Content’ fragment or

- indirectly the 'Schedule' fragment referring to the 'Service' fragment the 'Content' fragment is associated with.

Therefore the terminal SHALL first consider the 'Schedule' fragments associated directly with the 'Content' fragment and if none of them is applicable only then consider any of the indirectly associated 'Schedule' fragments.

## Appendix A. Change History

(Informative)

### A.1 Approved Version History

Reference	Date	Description
OMA-TS-BCAST_Service_Guide-V1_0-20090212-A	12 Feb 2009	Status changed to Approved by TP TP Ref # OMA-TP-2009-0071- INP_BCAST_V1_0_ERP_for_Notification_and_Final_Approval
OMA-TS-BCAST_Service_Guide-V1_0_1-20130109-A	09 Jan 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0001-INP_BCAST_V1_0_1_ERP_for_notification
OMA-TS-BCAST_Service_Guide-V1_1-20131029-A	29 Oct 2013	Status changed to Approved by TP TP Ref # OMA-TP-2013-0332-INP_BCAST_V1_1_ERP_for_final_Approval
OMA-TS-BCAST_Service_Guide-V1_2-20170131-A	31 Jan 2017	Status changed to Approved by TP TP Ref # OMA-TP-2017-0002-INP_BCAST-V1_2_ERP_for_Final_Approval

## Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

### B.1 SCR for BCAST Service Guide Client (SG-C)

Item	Function	Reference	Status	Requirement
BCAST-SG-C-001	Support for access to Interaction Channel	General	O	BCAST-SG-C-003 AND BCAST-SG-C-006 AND BCAST-SG-C-012 AND BCAST-SG-C-014 AND BCAST-SG-C-016
BCAST-SG-C-002	Support for Service Guide data model and XML Schema for Schedule, Content, Access, PurchaseItem, PurchaseData, PurchaseChannel and PreviewData fragments of Service Guide	Section 5.1	M	
BCAST-SG-C-003	Support for XML Schema for InteractivityData fragment of Service Guide	Section 5.1	O	
BCAST-SG-C-004	Display of all declared program guide elements or attributes in Service Guide fragments for presentation to end-users	Section 5.1.2	O	
BCAST-SG-C-005	Support for reception of preview data over Broadcast Channel	Section 5.1.2.9	M	
BCAST-SG-C-006	Support for reception of preview data over Interaction Channel	Section 5.1.2.9	O	
BCAST-SG-C-007	Support for split-TOI scheme	Section 5.4.1.1	O	
BCAST-SG-C-008	Support Service Guide encapsulation	Section 5.4.1.3	M	
BCAST-SG-C-009	Support GZIP for SGDU and SGDD compression	Section 5.4.1.4	M	
BCAST-SG-C-010	Support Service Guide Delivery Descriptor	Section 5.4.1.5	M	
BCAST-SG-C-011	Service Guide delivery over Broadcast Channel	Section 5.4.2	M	
BCAST-SG-C-012	Service Guide delivery over Interaction Channel	Section 5.4.3	O	
BCAST-SG-C-013	Support Service Guide update and management over Broadcast Channel	Section 5.5.1	M	

Item	Function	Reference	Status	Requirement
BCAST-SG-C-014	Support Service Guide update and management over Interaction Channel	Section 5.5.2	O	
BCAST-SG-C-015	Support initial Service Guide discovery over Broadcast Channel	Section 6.1.1	M	
BCAST-SG-C-016	Support initial Service Guide discovery over Interaction Channel	Section 6.1.2	O	
BCAST-SG-C-017	Support requests for specific Service Guide Delivery Descriptors	Section 5.4.3.3	O	
BCAST-SG-C-018	Support Service Guide discovery using DNS	Section 6.2.1	O	BCAST-SG-C-001
BCAST-SG-C-019	Support of Broadcast Terminal Provisioning signalling	Section 5.4.1.5	O	
BCAST-SG-C-020	Support RMS template retrieval signalling in SGDD	Section 5.4.1.5	O	
BCAST-SG-C-021	Support Terminal's location-based SG request	Section 5.4.3.4	O	
BCAST-SG-C-022	Support signalling of hybrid SG	Section 5.4.1.5.2	O	
BCAST-SG-C-023	Non-display of all SG content filtered-out, or display of all SG content filtered-in, by polarity in BroadcastArea location filters	Section 5.1.2.1, Section 5.1.2.3	O	
BCAST-SG-C-024	Support for inheritance of Service fragment fields by Content fragments (e.g. Popularity, Freshness, and BroadcastArea.)	Section 5.1.2.1, Section 5.1.2.3	O	

## B.2 SCR for BCAST Service Guide Generation/Adaptation/Distribution (SG-GAD)

Item	Function	Reference	Status	Requirement
BCAST-SGGAD-S-001	SGGAD-S serving BDS that is capable of supporting Broadcast Channel	General	O	BCAST-SGGAD-S-019 AND BCAST-SGGAD-S-021 AND BCAST-SGGAD-S-023
BCAST-SGGAD-S-002	SGGAD-S serving BDS that is capable of supporting Interaction Channel	General	O	BCAST-SGGAD-S-020 AND BCAST-SGGAD-S-022 AND BCAST-SGGAD-S-024
BCAST-SGGAD-S-003	Backend interface SG-2 exposed in implementation	Section 5.6	O	BCAST-SGGAD-S-004

Item	Function	Reference	Status	Requirement
BCAST-SGGAD-S-004	Support backend interface SG-2 for SG function	Section 5.3.1	O	BCAST-SGGAD-S-005
BCAST-SGGAD-S-005	Support Service Guide data model and the corresponding XML schema for the Service Guide data exchange	Section 5.1.2 , Section 5.3.1	O	
BCAST-SGGAD-S-006	Backend interface SG-4 exposed in implementation	Section 5.6	O	BCAST-SGGAD-S-007
BCAST-SGGAD-S-007	Support backend interface SG-4 for SG function	Section 5.6	O	(BCAST-SGGAD-S-008 OR BCAST-SGGAD-S-009) AND BCAST-SGGAD-S-010 AND (BCAST-SGGAD-S-011 OR BCAST-SGGAD-S-012) AND BCAST-SGGAD-S-013 AND BCAST-SGGAD-S-005
BCAST-SGGAD-S-008	Support IPv4	Section 5.6.2.1	O	
BCAST-SGGAD-S-009	Support IPv6	Section 5.6.2.1	O	
BCAST-SGGAD-S-010	Support TCP	Section 5.6.2.1	O	
BCAST-SGGAD-S-011	Support HTTP1.1	Section 5.6.2.1	O	
BCAST-SGGAD-S-012	Support HTTPS	Section 5.6.2.1	O	
BCAST-SGGAD-S-013	SG backend messages for content delivery	Section 5.6.2.2	O	
BCAST-SGGAD-S-014	Support announcing Service Guides within a service guide	Section 5.2	O	
BCAST-SGGAD-S-015	Support Service Guide fragment identification	Section 5.4.1.1	M	
BCAST-SGGAD-S-016	Support Service Guide encapsulation	Section 5.4.1.3	M	
BCAST-SGGAD-S-017	Support GZIP for SGDU and SGDD compression	Section 5.4.1.4	O	
BCAST-SGGAD-S-018	Support Service Guide Delivery Descriptor	Section 5.4.1.5	M	
BCAST-SGGAD-S-019	Service Guide delivery over Broadcast Channel	Section 5.4.2	O	
BCAST-SGGAD-S-020	Service Guide delivery over Interaction Channel	Section 5.4.3	O	
BCAST-SGGAD-S-021	Support Service Guide update and management over Broadcast Channel	Section 5.5.1	O	
BCAST-SGGAD-S-022	Support Service Guide update and management over Interaction Channel	Section 5.5.2	O	
BCAST-SGGAD-S-023	Support delivery of preview data over Broadcast Channel	Section 5.1.2.9	O	



Item	Function	Reference	Status	Requirement
BCAST-SGGAD-S-024	Support delivery of preview data over Interaction Channel	Section 5.1.2.9	O	
BCAST-SGGAD-S-025	Support Service Guide discovery using DNS	Section 6.2.1	O	BCAST-SGGAD-S-002
BCAST-SGGAD-S-026	Support of Broadcast Terminal Provisioning signalling	Section 5.4.1.5	O	
BCAST-SGGAD-S-027	Support for RMS SG signalling	Section 5.4.1.5	O	
BCAST-SGGAD-S-028	Support terminal's BDSLocationID-based SG request	Section 5.4.3.4	O	
BCAST-SGGAD-S-029	Support signalling of hybrid SG	Section 5.4.1.5.2	O	

### B.3 SCR for BCAST Service Guide Subscription Source (SGSS)

Item	Function	Reference	Status	Requirement
BCAST-SGSS-S-001	Backend interface SG-4 exposed in implementation	Section 5.6	O	BCAST-SGSS-S-002
BCAST-SGSS-S-002	Support backend interface SG-4 for SG function	Section 5.6	O	(BCAST-SGSS-S-003 OR BCAST-SGSS-S-004) AND BCAST-SGSS-S-005 AND (BCAST-SGSS-S-006 OR BCAST-SGSS-S-007) AND BCAST-SGSS-S-008 AND BCAST-SGSS-S-009
BCAST-SGSS-S-003	Support IPv4	Section 5.6.2.1	O	
BCAST-SGSS-S-004	Support IPv6	Section 5.6.2.1	O	
BCAST-SGSS-S-005	Support TCP	Section 5.6.2.1	O	
BCAST-SGSS-S-006	Support HTTP1.1	Section 5.6.2.1	O	
BCAST-SGSS-S-007	Support HTTPS	Section 5.6.2.1	O	
BCAST-SGSS-S-008	SG backend message for content delivery	Section 5.6.2.2	O	
BCAST-SGSS-S-009	Support Service Guide data model and the corresponding XML schema for the Service Guide data exchange	Section 5.3.1	O	

### B.4 SCR for BCAST Service Guide Application Source (SGAS)

Item	Function	Reference	Status	Requirement
BCAST-SGAS-S-001	Backend interface SG-1 exposed in implementation	Section 5.6	O	BCAST-SGAS-S-002
BCAST-SGAS-S-002	Support backend interface SG-1 for SG function	Section 5.3.1	O	

Item	Function	Reference	Status	Requirement
BCAST-SGAS-S-003	Backend interface SG-2 exposed in implementation	Section 5.6	O	BCAST-SGAS-S-004
BCAST-SGAS-S-004	Support backend interface SG-2 for SG function	Section 5.3.1	O	BCAST-SGAS-S-005
BCAST-SGAS-S-005	Support Service Guide data model and the corresponding XML schema for the Service Guide data exchange	Section 5.3.1	O	

## Appendix C. Service Guide Application Scenarios (Informative)

### C.1 Scenario of selective viewing based on the codec requirement in the SG

This application scenario describes how the same content using different codec can be subscribed and consumed by the terminals with different AV Codec capabilities via different channels.

Assuming there is a video streaming program in two different formats, MPEG-2 and H.264. To facilitate the terminals that support H.264 to access the high quality video stream, and the terminals that only support MPEG-2 to access the low quality video stream, the following steps may be used.

Step1. SG provider creates one 'Service' fragment for the video streaming program that is encoded in two different formats MPEG-2 and H.264.

Step 2. SG provider creates two 'Access' fragments associated with the above 'Service' fragment to respectively specify the two access channels for the two video stream formats. In the "TerminalCapabilityRequirement" element of every 'Access' fragment, the video codec support requirement is explicitly specified.

Step 3. After receiving and reading the Service Guide, the terminal or the user selects the preferred access channel based on the terminal's AV decoding capabilities. Then the terminal accesses to the selected channel which delivers the corresponding video stream. Please note, if the service is not free, some necessary subscription procedures will be involved.

### C.2 Scenario of selective viewing based on the bandwidth requirement in the SG

This application scenario describes how the SG can support the selective viewing of the different versions (for example, basic version only contains audio; normal version contains both audio and video; or the basic version contains the low bit rate stream of the live show, but the normal version contains the high bit rate stream of the same live show) of the same real-time program with different bandwidth requirements. The selective viewing provides more flexibility to the terminal users and ensures the users can always consume their interested program even as the terminal is under a bad reception condition, consequently enhances the user experience.

Assuming there are two versions of the same streaming program, the basic version requires 200kbps for the bandwidth and the normal version requires 500kbps as the bandwidth. To facilitate the terminals that are under good reception condition (>500kbps) to access the normal version of the program, and the terminals that are under the bad reception condition (<200kbps) to access the basic version of the program, the following steps may be used.

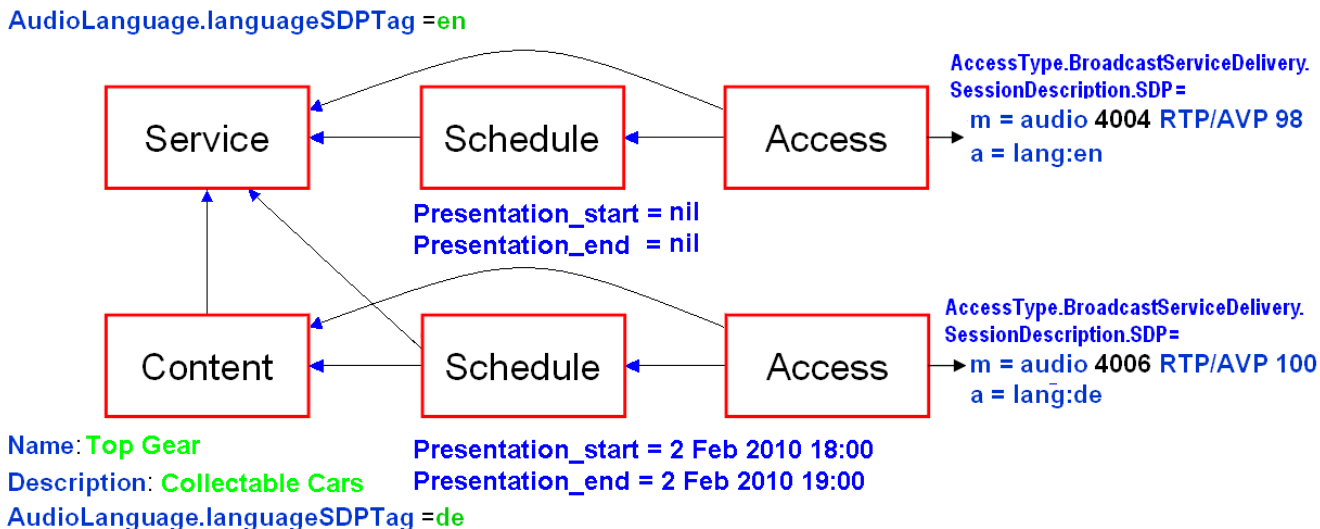
Step1. SG provider creates a 'Service' fragment for the streaming program.

Step 2. SG provider creates two 'Access' fragments associated with the above 'Service' fragment to respectively specify the two access channels, one access only delivers the basic version of the live show which only contains the audio component or contains the low bit rate streams of the original audio and video streams, another access delivers the normal version of the live show which contains the original high bit rate streams of the audio and video streams. In the "BandwidthRequirement" element of every 'Access' fragment, the bandwidth requirement for every access is explicitly specified.

Step 3. As the program is broadcast via two access channels, the terminal or the user selects the appropriate access channel based on the current reception condition of the terminal. If the terminal reception condition can not meet the bandwidth requirement of the normal version of the access, then the terminal will listen to basic access channel. Otherwise, the terminal will be able to access to the normal access channel.

### C.3 Scenario of a TV program which is broadcasted in two languages.

In this scenario a TV show on a certain TV channel is broadcasted in two different languages.



**Figure 15: Use of ‘Service’, ‘Content’, ‘Schedule’ and ‘Access’ fragment to declare a service with a TV program in two different languages.**

In this example, the language is specified with ‘languageSDPTag’s. The show “Top Gear” is presented in English and in German. The Default Access fragment declares that the Service is available for an unlimited time period with an AudioLanguage.languageSDPTag=en, i.e. the service by default is available in English. An alternative language, German (de), is declared only for the ‘Content’ fragment for a particular show, “Top Gear” (through AudioLanguage.languageSDPTag) and the ‘Access’ fragment for that ‘Content’ fragment declares a different source port number for the second audio stream.

TV programs with a second language can be also useful, for roaming users. They might be provided with an ‘Access’ fragment pointing to a UMTS stream, which gives them the audio comment in their local language via the interactive channel. The video content could still be the same broadcast stream for these roaming users. These ESG fragments for roaming users could be either distributed via the back-end systems over the broadcast bearer or retrieved via the interactive channel.

## C.4 Scenario of a broadcast streaming service protected by multiple Key Management Systems

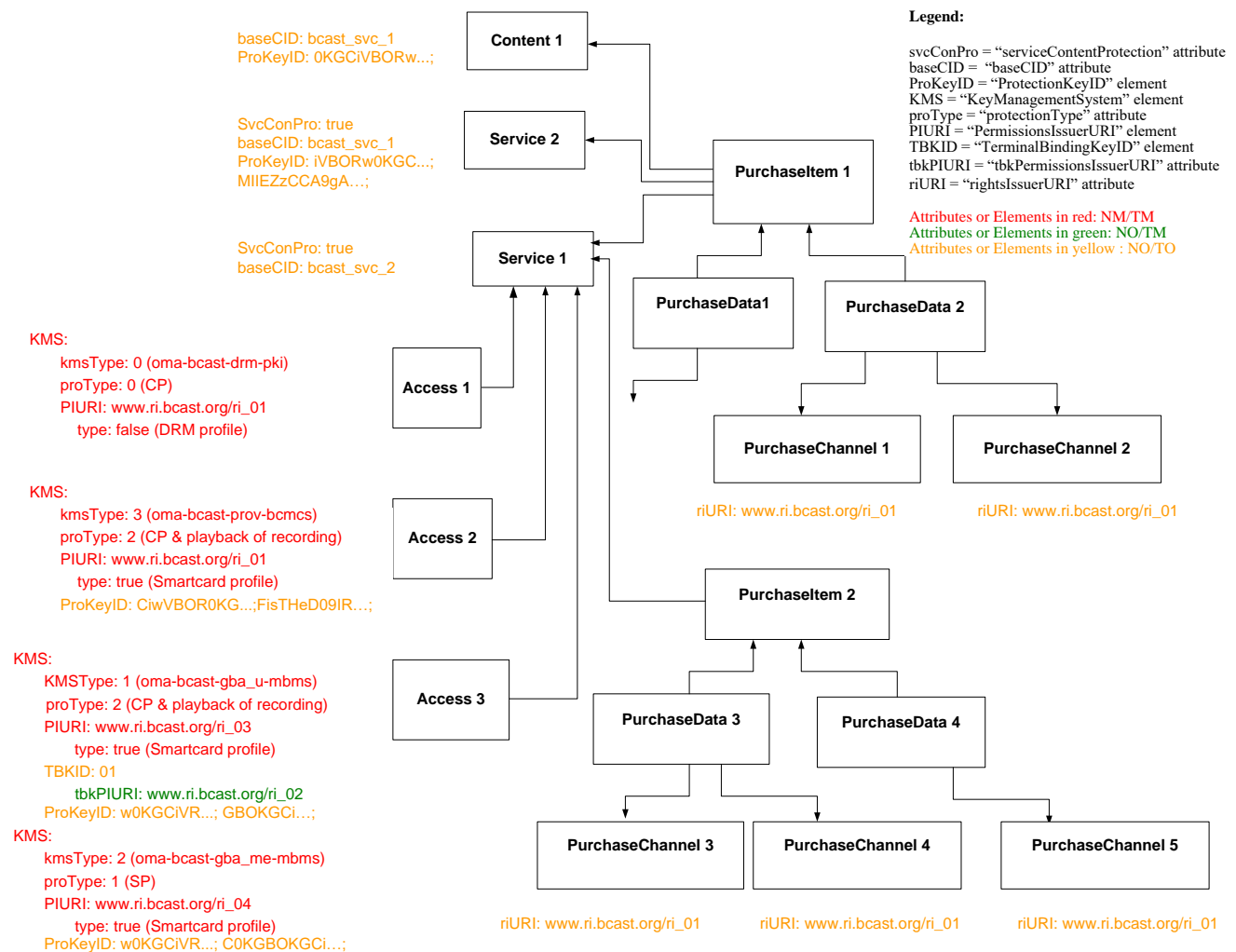


Figure 16: Use of SG to signal service and content protection information

This application scenario describes how the service and content protection information of a broadcast service protected by multiple Key Management Systems (KMS) is signalled using the Service Guide.

In the example shown in Figure 16, to access the protected Service 1 there are three accesses specified by three ‘Access’ fragments, Access 1, Access 2 and Access 3, for each of which different KMS are used. The KMS related information is signalled by ‘KeyManagementSystem’ element in the ‘Access’ fragment. From the linkage between the ‘Service’ fragment and the ‘Access’ fragments, the terminal can easily find out by how many and what exact KMSs a specific service is protected.

In this example, the KMS type of Access 1 is “oma-bcast-drm-pki”, the protection type used with this KMS is Content Protection as defined by LTKM (i.e. protection\_after\_reception in STKM = 0x00 or 0x01 [BCAST12-ServContProt]); the Permissions Issuer URI to acquire SEK/PEK is “www.ri.bcast.org/ri\_01”, and in case of Access 1 the Permissions Issuer is synonymous with the Rights Issuer because DRM Profile is used here.

The KMS type of Access 2 is “oma-bcast-prov-bcmcs”, the protection type used with this KMS is Content Protection as defined by LTKM, plus playback of protected recording permission (i.e. protection\_after\_reception in STKM = 0x02); The

Permissions Issuer URI to acquire SEK/PEK is also “www.ri.bcast.org/ri\_01”, identical to that for Access 1, and which means one Permissions Issuer can support multiple KMSs.

For Access 3 two types of KMS are supported; one is “oma-bcast-gba\_u-mbms”, the other is “oma-bcast-gba\_me-mbms”. The protection type of the former is Content Protection as defined by LTKM, plus playback of protected recording permission (i.e. protection\_after\_reception in STKM = 0x02). Protection type of the latter KMS is Service Protection only (i.e. protection\_after\_reception in STKM = 0x03). The Permissions Issuer URIs are “www.ri.bcast.org/ri\_03” and “www.ri.bcast.org/ri\_04” respectively.

Besides the above service and content information which are common to both DRM profile and Smartcard profile, some other parameters in SG are only specific to one profile. Those parameters are only required to be supported when the relevant profile is supported.

For the DRM Profile, part of the Service or Program CID used to identify the corresponding asset within a OMA DRM 2.0 Rights Object are signalled as ‘baseCID’ attribute in the corresponding ‘Service’ fragment and ‘Content’ fragment. The Service or Program CID is obtained from the BaseCID as described in [BCAST12-ServContProt]. Furthermore, to inform the terminal which Rights Issuer may provide the Rights Issuer Services [DRM20-Broadcast-Extensions] operated by its home BSM, a ‘rightsIssuerURI’ attribute may also be signalled in the ‘PurchaseChannel’ fragment in case of DRM profile.

For the Smartcard profile, the list of the key identifiers needed to access the protected service or content are signalled as ‘ProtectionKeyID’ element in ‘Service’, ‘Content’, ‘Access’ or ‘PurchaseItem’ fragment so that the terminal can determine whether or not it has the correct key material to access the service or content. In the above example, the key identifiers for Content 1 and Service 2 are indicated in the fragment itself because they are not shared by multiple service providers. However, for Service 1, because multiple means of access, associated with different KMS(s) are provided (possibly each from a different service provider), the protection keys in each KMS are different. In this case the protection key identifiers are signalled under ‘KeyManagementSystem’ element of each ‘Access’ fragment, rather than in the ‘Service’ fragment. Furthermore, in the case of Smartcard Profile, Terminal Binding Key (TBK) can be used to bind the broadcast service to valid terminals for content protection. The identifier of the TBK is signalled as ‘TerminalBindingKeyID’ sub-element of ‘KeyManagementSystem’. The GBA\_U KMS of Access 3 in the above example uses TBK. Because the Permissions Issuer from which the TBK is acquired may be different than the Permissions Issuer from which the SEK/PEK is acquired, a TBK specific Permissions Issuer is specified as ‘tbkPermissionsIssuerURI’ attribute under ‘TerminalBindingKeyID’ element. However in some cases the same Permissions Issuer is used to provide both SEK/PEK and TBK, and for which the ‘tbkPermissionsIssuerURI’ attribute will not be instantiated.

## C.5 Scenario of a main service with interaction

This application scenario describes how the Service Guide can be used to describe the scenario in which the main service (for example, mobile TV) is enhanced with interaction (for example, SMS voting). Figure 17 depicts the scenario on a timeline. Figure 18 depicts the necessary Service Guide fragment and their relation. Figure 19 depicts the example encapsulation to SGDU. Note: in this example case the schedule of content is the same as the delivery schedule of InteractivityMediaDocuments. If different schedule would be needed, can one achieve that by instantiating two ‘Schedule’ fragments – one for Content and one for the delivery of Interactive Media Documents.

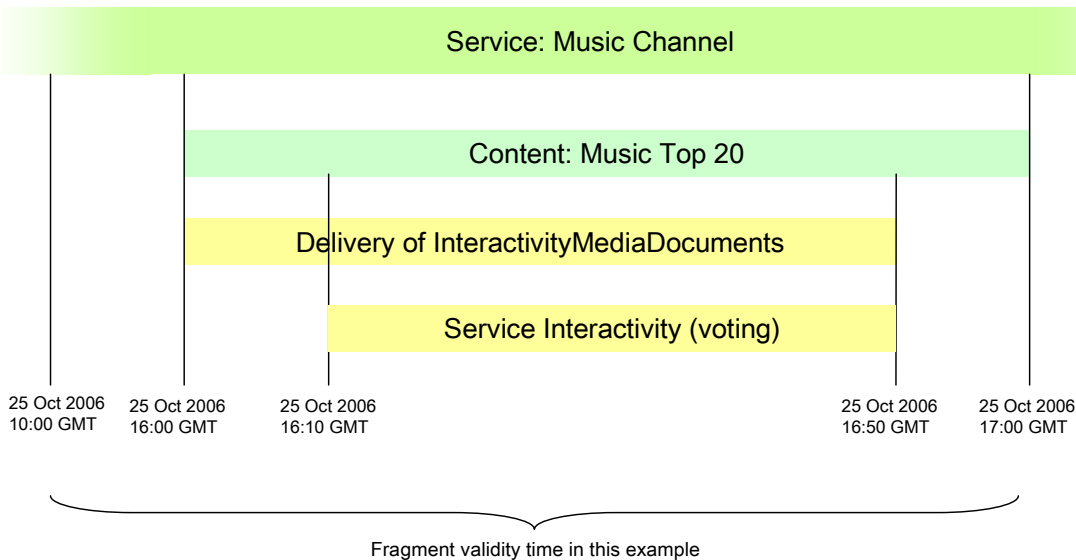


Figure 17: Service example on a timeline

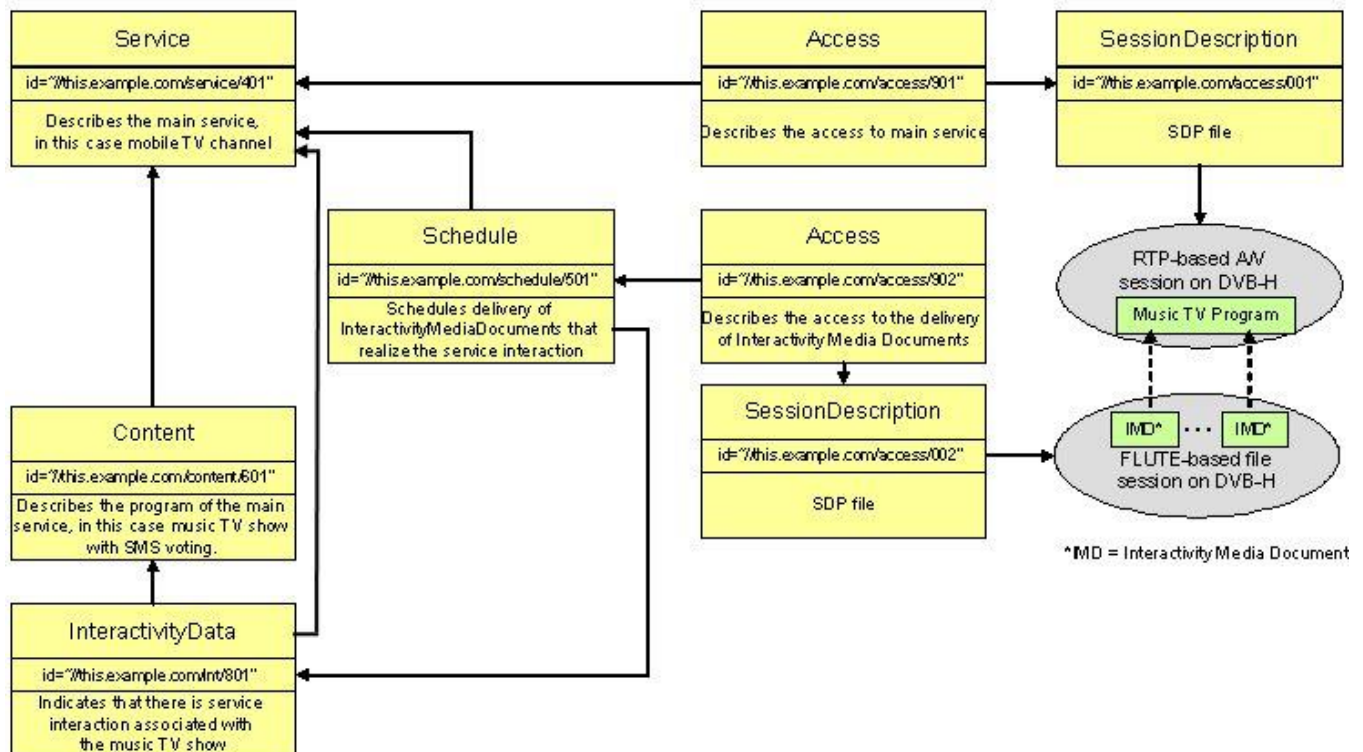


Figure 18: Service Guide example for main service with interactivity

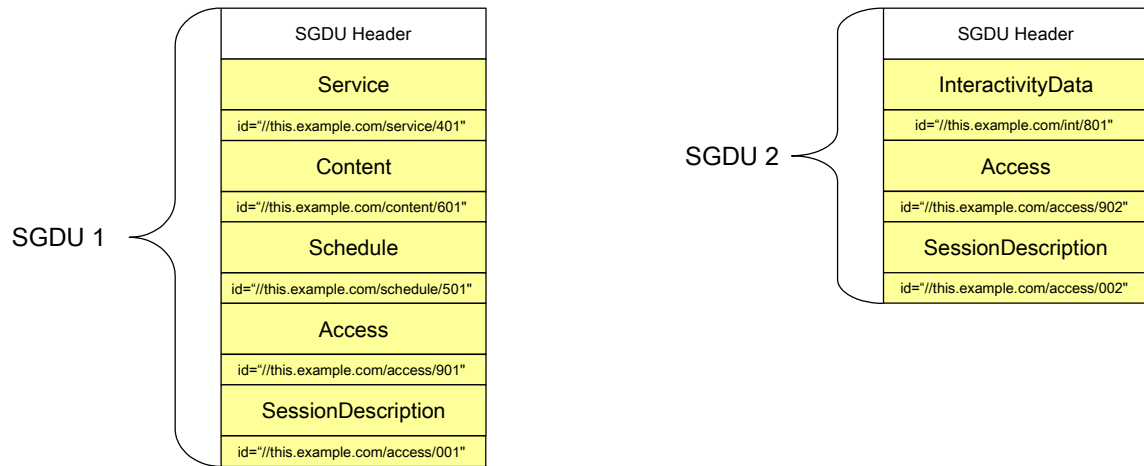


Figure 19: Simple encapsulation of Service Guide fragments in this example

The following sections give example instantiations of the Service Guide fragments in this example.

### C.5.1 Service (//this.example.com/service/401)

Name	Type	Example	Data Type
Service	E		
id	A	//this.example.com/service/401	anyURI
version	A	1	unsignedInt
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt
weight	A	2	unsignedInt
ServiceType	E1	1 (Basic TV)	unsigned Byte
<b>Start of program guide</b>			
Name	E1	Music Channel (xml:lang="en")	string
Name	E1	Musiikkikanava (xml:lang="fin")	string
Genre	E1	Music	string
Extension	E1		
url	A	http://this.example.com/music_channel.html	anyURI
Description	E2	Home page of this service (xml:lang="en")	string
Description	E2	Kanavan kotisivu (xml:lang="fin")	string
<b>End of program guide</b>			

### C.5.2 Schedule (//this.example.com/schedule/501)

Name	Type	Example	Data Type
Schedule	E		
id	A	//this.example.com/schedule/501	anyURI
version	A	5	unsignedInt
defaultSchedule	A	true	boolean
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt



ServiceReference	E1		
idRef	A	//this.example/service/401	anyURI
InteractivityDataReference	E1		
idRef	A	//this.example.com/int/801	anyURI
DistributionWindow	E2		
startTime	A	25 Oct 2006, 16:00 GMT in NTP	unsignedInt
endTime	A	25 Oct 2006, 16:50 GMT in NTP	unsignedInt

### C.5.3 Content (//this.example.com/content/601)

Name	Type	Example	Data Type
Content	E		
id	A	//this.example.com/content/601	anyURI
version	A	10	unsignedInt
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt
Type	A	"video"	string
ServiceReference	E1		
idRef	A	//this.example.com/service/401	anyURI
<b>Start of program guide</b>			
Name	E1	Music Top 20 (xml:lang="en")	String
StartTime	E1	2006-10-25T16:00:00Z	dateTime
EndTime	E1	2006-10-25T17:00:00Z	dateTime
Genre	E1	Music	string
<b>End of program guide</b>			

### C.5.4 InteractivityData (//this.example.com/int/801)

Name	Type	Example	Data Type
InteractivityData	E		
id	A	//this.example.com/int/801	anyURI
version	A	2	unsignedInt
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt
preListenIndicator	A	false	boolean
interactivityMediaDocumentPointer	A	1881rt45tqw	anyURI
InteractivityType	E1	"voting"	string
ServiceReference	E1		
idRef	A	//this.example.com/service/401	anyURI
ContentReference	E1		
idRef	A	//this.example.com/content/601	anyURI
InteractivityWindow	E1		
startTime	A	25 Oct 2006, 16:10 GMT in NTP	unsignedInt
endTime	A	25 Oct 2006, 16:50 GMT in NTP	unsignedInt
BackOffTiming	E1		
offsetTime	A	3	decimal

randomTime	A	2	decimal
------------	---	---	---------

### C.5.5 Access (//this.example.com/access/901)

Name	Type	Example	Data Type
Access	E		
id	A	//this.example.com/access/901	anyURI
version	A	2	unsignedInt
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt
AccessType	E1		
BroadcastServiceDelivery	E2		
BDSType	E3	0 (IPDC over DVB-H)	unsignedByte
Version	E4	1	string
SessionDescriptionReference	E3		
type	A	0 – SDP	unsignedByte
idRef	A	bcast://this.operator/access/001	anyURI
ServiceReference	E1		
idRef		//this.example.com/service/401	anyURI
ServiceClass	E1	urn:oma:bcast:oma_bsc:st:1.0	string

### C.5.6 Access (//this.example.com/access/902)

Name	Type	Example	Data Type
Access	E		
id	A	//this.example.com/access/902	anyURI
version	A	8	unsignedInt
validFrom	A	25 Oct 2006, 10:00 GMT in NTP	unsignedInt
validTo	A	25 Oct 2006, 17:00 GMT in NTP	unsignedInt
AccessType	E1		
BroadcastServiceDelivery	E2		
BDSType	E3	0. IPDC over DVB-H	unsignedByte
Version	E4	1	string
SessionDescriptionReference	E3		
type	A	0 – SDP	unsignedByte
idRef	A	//this.example.com/access/002	anyURI
ScheduleReference	E1		
idRef	A	//this.example.com/schedule/502	anyURI
ServiceClass	E1	urn:oma:bcast:oma_bsc:fc:1.0	String

## C.6 Scenario of a main service with purchasable protected auxiliary data content

This application scenario describes how the Service Guide can be used to describe the scenario in which the main service (for example, mobile TV) is enhanced with purchasable, protected auxiliary data (for example, sports statistics related to sports event main service). In this case, the main service is provided free to air. Figure 20 depicts the scenario on a timeline. Figure 21 depicts the necessary Service Guide fragments and their relations. Figure 22 depicts the example encapsulation to SGDU.

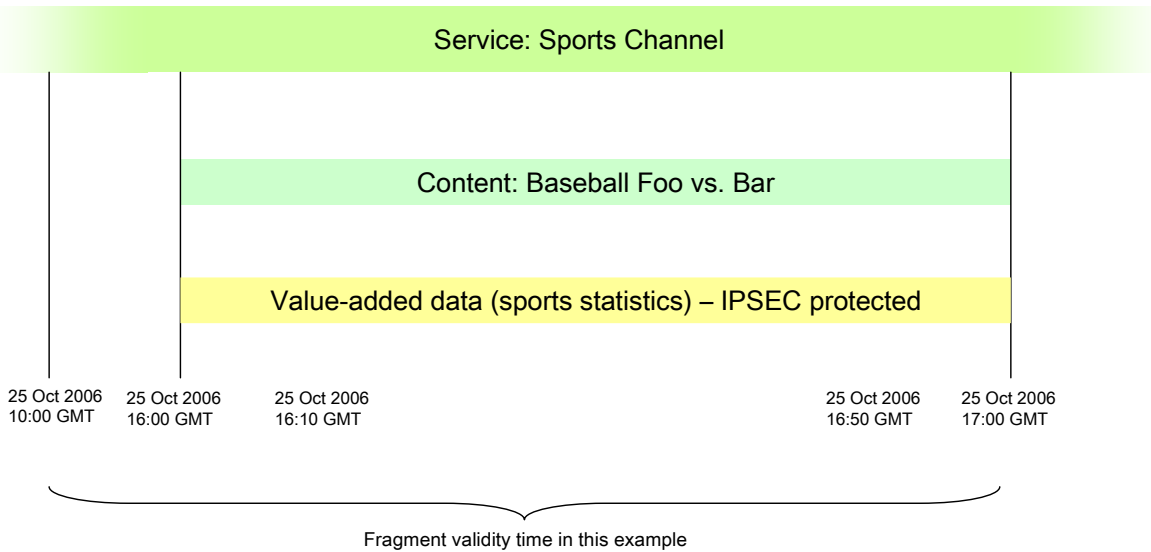


Figure 20: Service example on a timeline

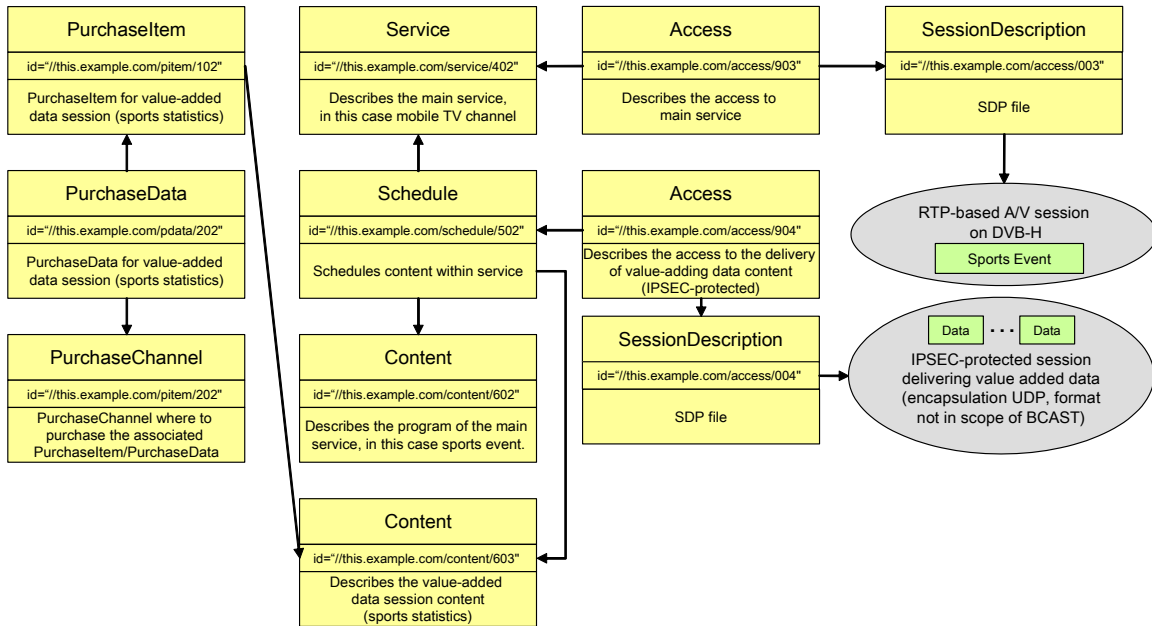


Figure 21: Service Guide example for main Service with value-added, protected data

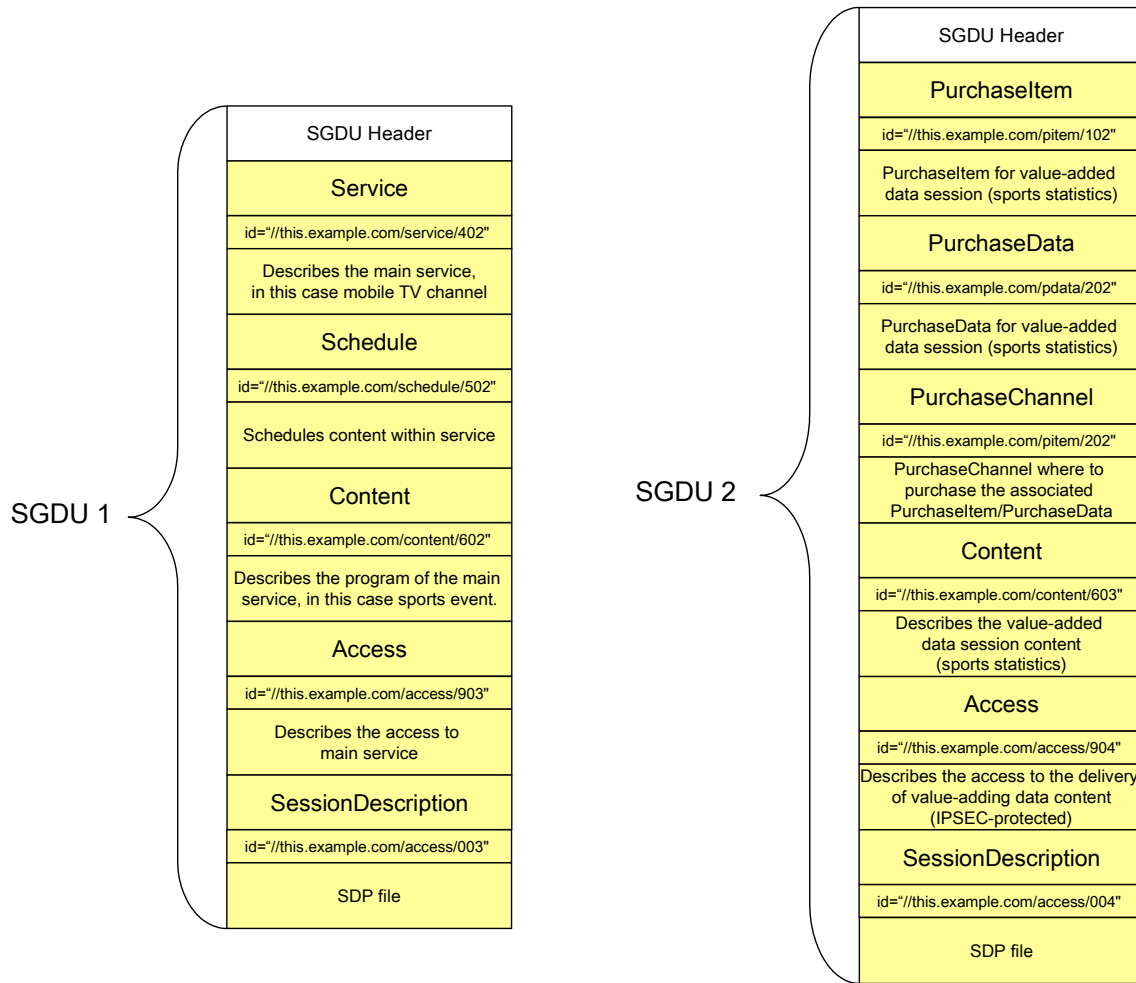


Figure 22: Simple encapsulation of Service Guide fragments in this example

## C.7 Scenario of multiple services provided with single channel each

This application scenario describes how the Service Guide can be used to describe the scenario in which multiple services provided with single channel each. Figure 23 depicts the necessary Service Guide fragments and their relations. Figure 24 depicts the example encapsulation to SGDUs.

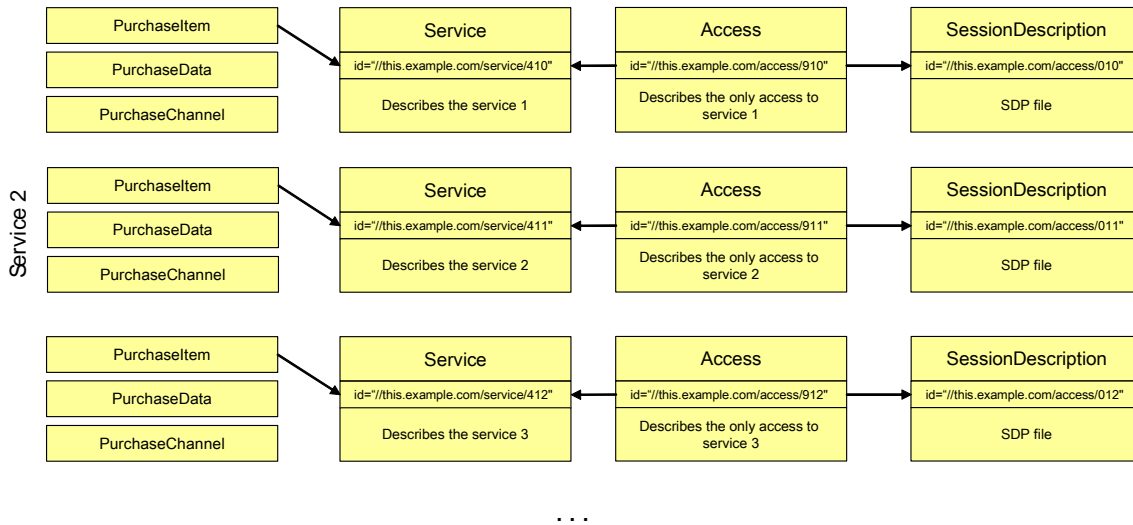


Figure 23: Service Guide example

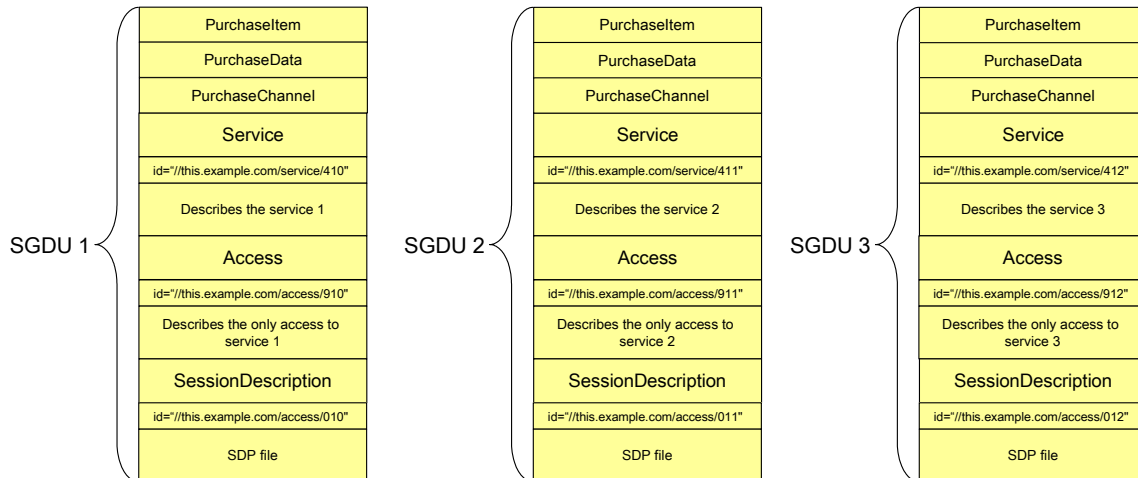


Figure 24: Simple encapsulation of Service Guide fragments in this example

## C.8 Scenario of bundling multiple channels into a single service

This application scenario describes how the Service Guide can be used to describe the scenario in which multiple channels are bundled with single service. In practice, each channel is represented as service and bundling is done through PurchaseItem. Figure 25 depicts the necessary Service Guide fragments and their relations. Figure 26 depicts the example encapsulation to SGDUs.

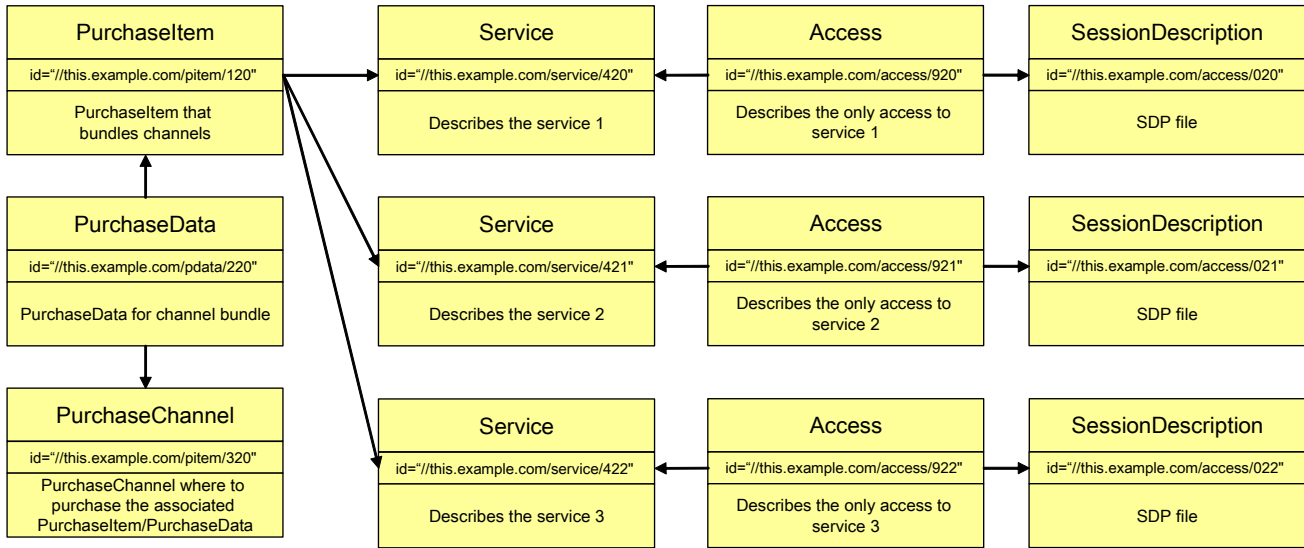


Figure 25: Service Guide figure

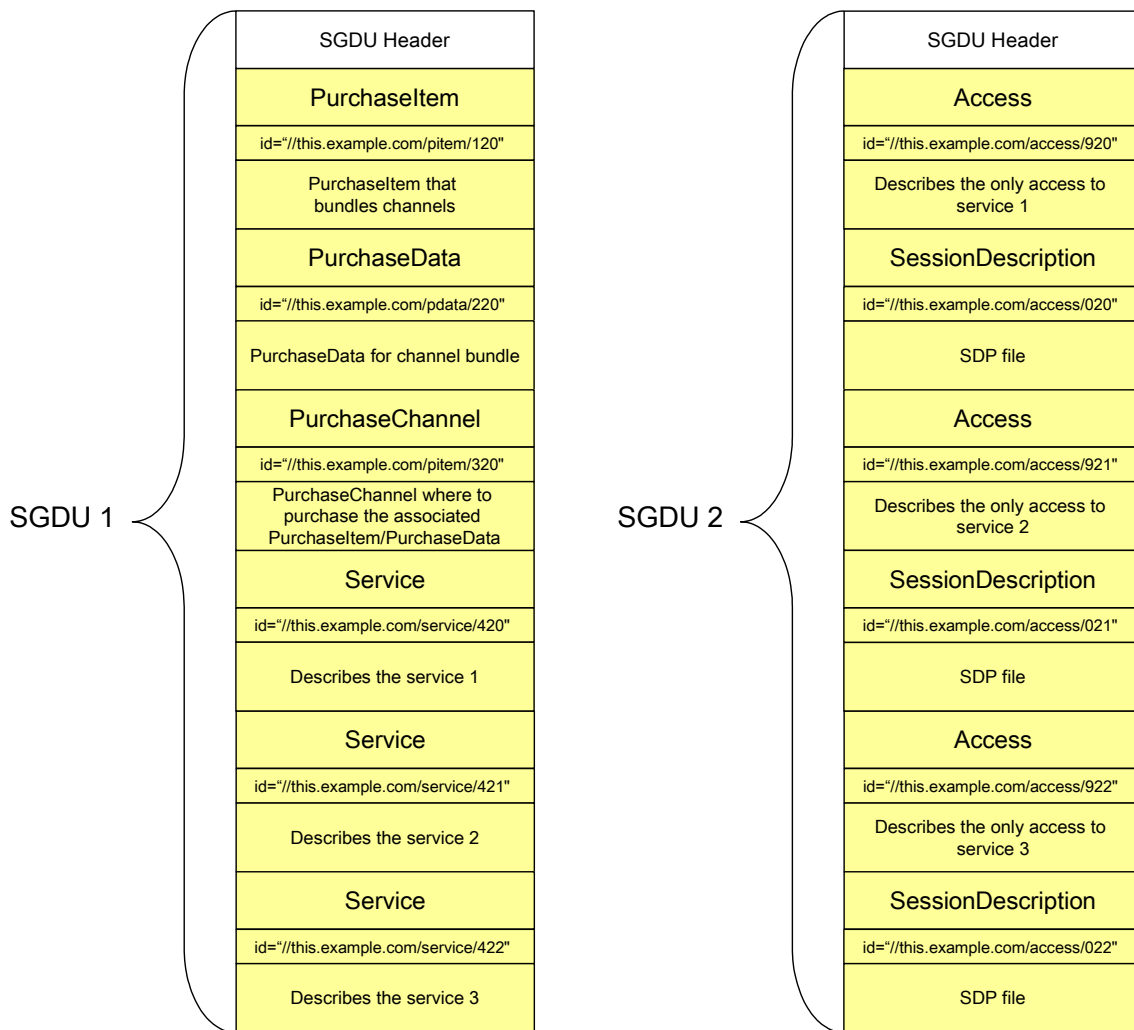


Figure 26: Simple encapsulation of Service Guide fragments in this example

### C.9 Scenario of using single Service Guide to provide service descriptions for multiple service providers

This application scenario describes how single Service Guide provides separate service descriptions for multiple service providers. In this case the main Service Guide is an aggregate of individual service descriptions, each associated with an individual Service Provider. Figure 27 depicts the necessary Service Guide fragments and their relations. In this example, two Service Providers (X and Y) are providing Service Guide fragments within the same Service Guide.

The association between the service providers and the individual fragments is provided using the grouping method of SGDD. The example grouping applying to this example case is provided in Tables 6 and 7. The eight fragments of the example are thought to be delivered using two SGDUs having TOIs two and nine. The SGDU having TOI two contains the fragments associated with the operator X and the SGDU having TOI nine contains the fragments associated with operator Y. The exception is ‘Service’ fragment having id //this.example.com/service/431. That fragment is associated with both the operators X and Y being delivered in the SGDU having TOI two. These eight fragments are in turn declared using two SGDDs, one representing the fragments associated with the operator X and the other representing the fragments associated with operator Y. Both the SGDDs also represent the subset of the Service Guide representing information for time interval from one to two P.M. on the 20<sup>th</sup> of December 2006. This is achieved using two grouping criteria, namely the time grouping criteria and BSM selector criteria, simultaneously.

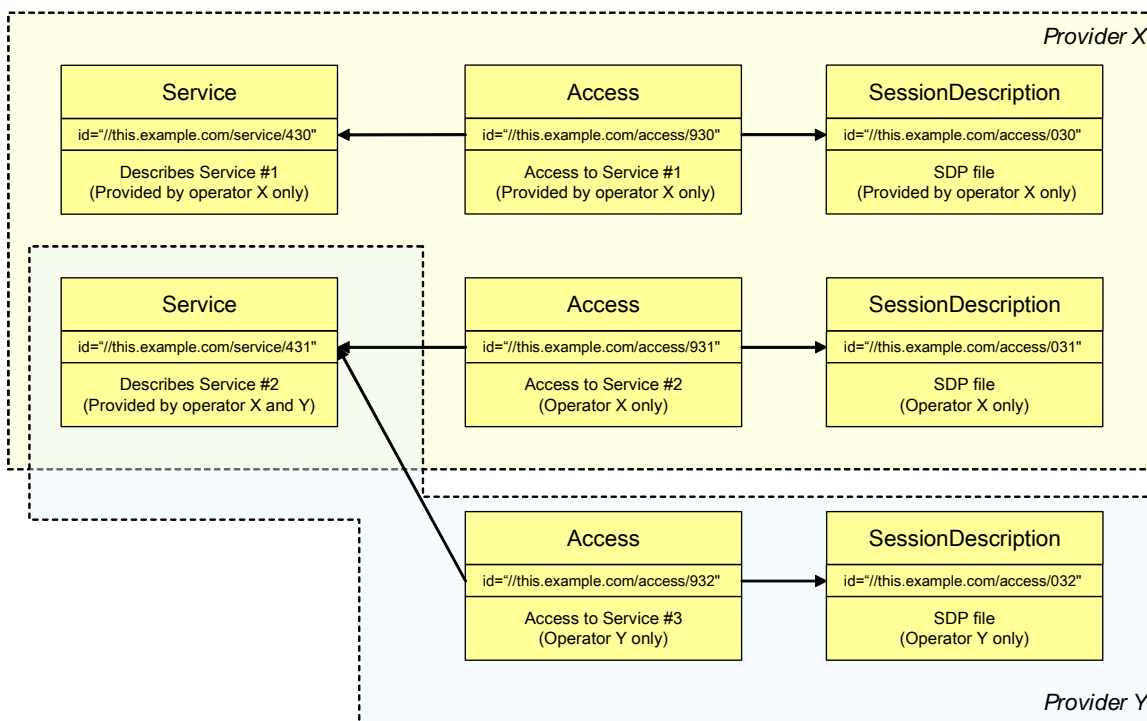


Figure 27: Service Guide figure

Name	Type	Example	Data Type
ServiceGuideDeliveryDescriptor	E		
id	A	//bsda.com/sgdd/1234	anyURI
version	A	3	unsignedInt
NotificationReception	E1		

port	A	34343	unsignedInt
address	A	232.0.0.1	string
DescriptorEntry	E1		
GroupingCriteria	E2		
TimeGroupingCriteria	E3		
startTime	A	20 Dec 2006, 13:00 GMT in NTP	unsignedInt
endTime	A	20 Dec 2006, 14:00 GMT in NTP	unsignedInt
BSMSelector	E3		
id	A	operatorX.com	anyURI
BSMFilterCode	E4	OPERATOR_X	string
type	A	1	unsignedByte
Name	E4	Operator X	string
Transport	E2		
ipAddress	A	232.0.0.2	string
port	A	8765	unsignedShort
srcIpAddress	A	1.2.3.4	string
transmissionSessionID	A	456	unsignedShort
ServiceGuideDeliveryUnit	E2		
transportObjectID	A	2	unsignedInt
validFrom	A	20 Dec 2006, 13:30 GMT in NTP	
validTo	A	20 Dec 2006, 14:00 GMT in NTP	
Fragment	E3		
transportID	A	1001	unsignedInt
id	A	//this.example.com/service/430	anyURI
version	A	4	unsignedInt
Fragment	E3		
transportID	A	1002	unsignedInt
id	A	//this.operatorexample.com/access/930	anyURI
Version	A	2	unsignedInt
Fragment	E3		
transportID	A	1003	unsignedInt
id	A	//this.example.com/access/030	anyURI
Version	A	6	unsignedInt
Fragment	E3		
transportID	A	1004	unsignedInt
id	A	//this.example.com/service/431	anyURI
version	A	2	unsignedInt
transportID	A	1005	unsignedInt
id	A	//this.example.com/access/931	anyURI
version	A	1	unsignedInt
Fragment	E3		
transportID	A	1006	unsignedInt
id	A	//this.example.com/access/031	anyURI
version	A	3	unsignedInt



Table 9: SGDD Example

Name	Type	Example	Data Type
ServiceGuideDeliveryDescriptor	E		
id	A	//bsda.com/sgdd/4567	anyURI
version	A	2	unsignedInt
NotificationReception	E1		
port	A	34343	unsignedInt
address	A	232.0.0.1	string
DescriptorEntry	E1		
GroupingCriteria	E2		
TimeGroupingCriteria	E3		
startTime	A	20 Dec 2006, 13:00 GMT in NTP	unsignedInt
endTime	A	20 Dec 2006, 14:00 GMT in NTP	unsignedInt
BSMSelector	E3		
id	A	//operatorY.com	anyURI
BSMFilterCode	E4	OPERATOR_Y	string
type	A	1	unsignedByte
Name	E4	Operator Y	string
Transport	E2		
ipAddress	A	232.0.0.2	string
port	A	8765	unsignedShort
srcIpAddress	A	1.2.3.4	string
transmissionSessionID	A	456	unsignedShort
ServiceGuideDeliveryUnit	E2		
transportObjectID	A	2	unsignedInt
validFrom	A	20 Dec 2006, 13:30 GMT in NTP	
validTo	A	20 Dec 2006, 14:00 GMT in NTP	
Fragment	E3		
transported	A	1004	unsignedInt
id	A	//this.example.com/service/431	anyURI
version	A	2	unsignedInt
ServiceGuideDeliveryUnit	E2		
transportObjectID	A	9	unsignedInt
validFrom	A	20 Dec 2006, 13:30 GMT in NTP	
validTo	A	20 Dec 2006, 14:00 GMT in NTP	
Fragment	E3		
transportID	A	1007	unsignedInt
id	A	//this.example.com/access/932	anyURI
version	A	9	unsignedInt
Fragment	E3		
transportID	A	1008	unsignedInt
id	A	//this.example.com/access/032	anyURI

version	A	1	unsignedInt
---------	---	---	-------------

Table 10: SGDD Example

## C.10 Scenario of Service Guide announcing other Service Guides

This application scenario describes how single Service Guide can refer to other Service Guides (for example, to provide Service Guides for multiple service providers). In this case the main Service Guide is an aggregate of individual Service Guides that are referred from the top level Service Guide. Provisions of section 5.2 (“Announcing Service Guides within a Service Guide”) can be applied. Figure 28 depicts the necessary Service Guide fragments and their relations. Figure 29 depicts the example encapsulation to SGDUs.

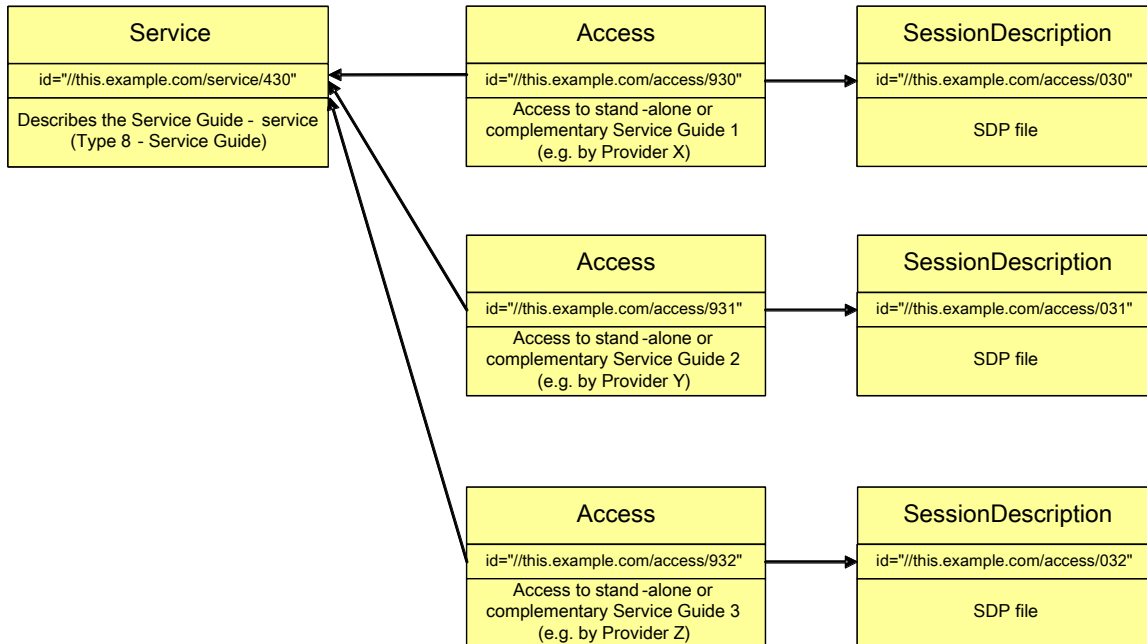


Figure 28: Service Guide example

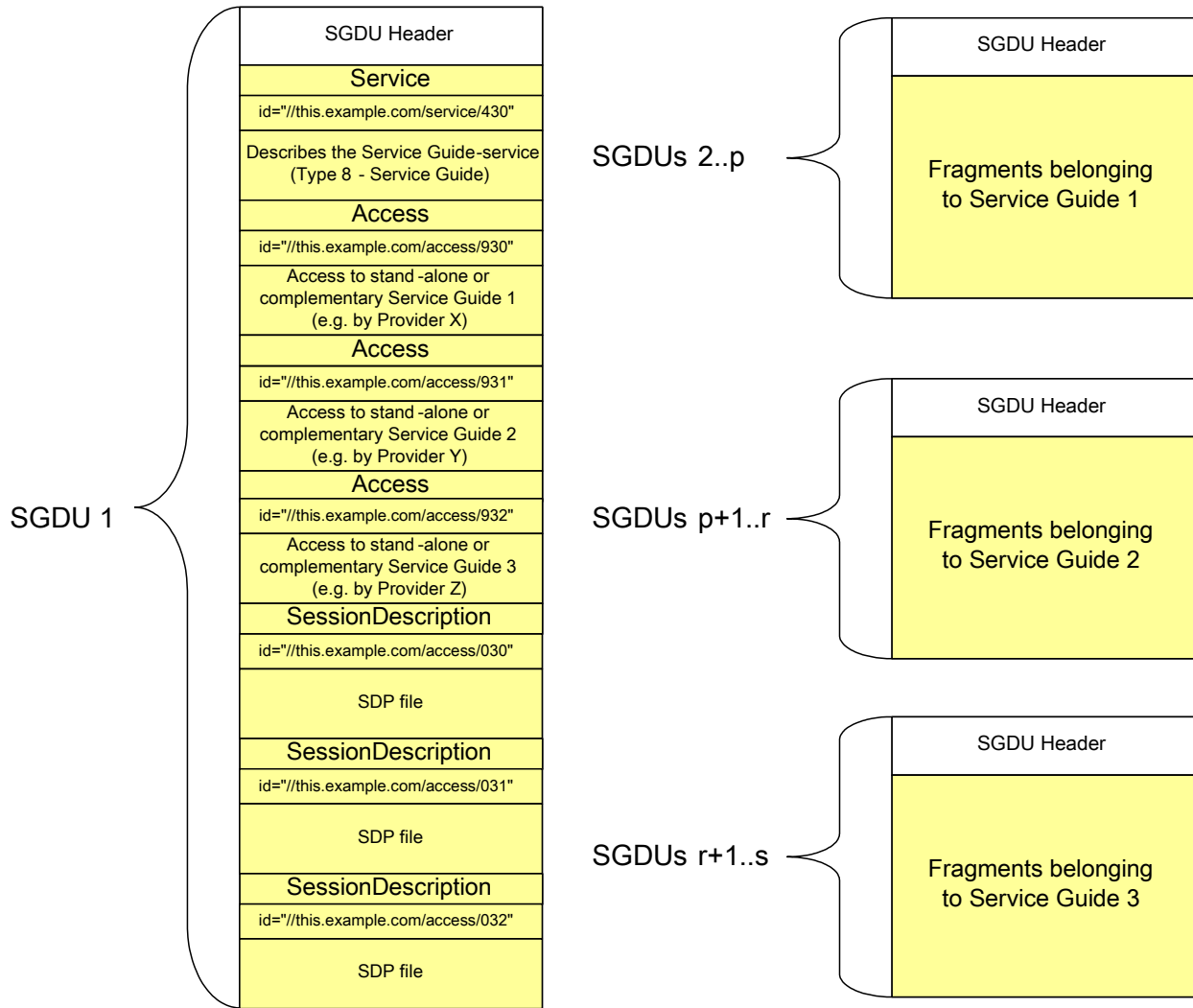


Figure 29: Simple encapsulation of Service Guide fragments in this example

## C.11 Scenario of FLUTE File Download Service with File Descriptions in the Service Guide

Files delivered in a file download service can be described in the Service Guide through dedicated 'Content' fragments so that the user has descriptive information about the content of each of those files. The application scenario in this section explains one example of how the Service Guide can be instantiated to include file descriptions and thereby enabling the possibility for the terminal to display files delivered in a FLUTE session without joining it. See section I.3.3 for a Service Guide instantiation where the files are only described in the FDT of the FLUTE session.

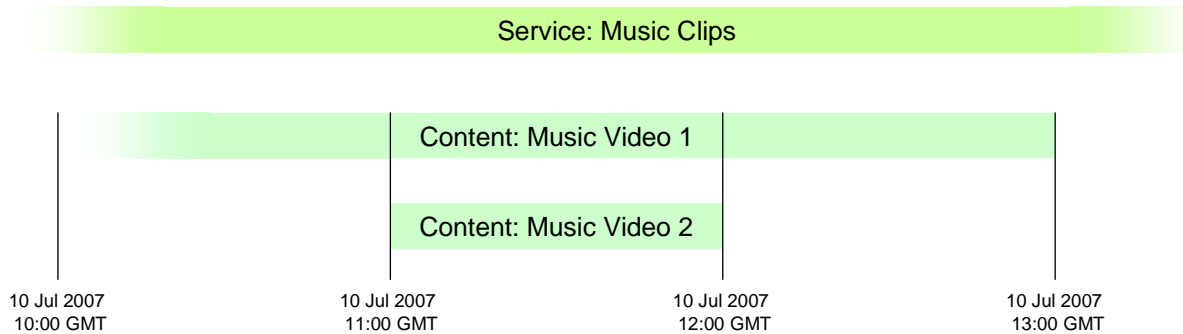


Figure 30: Service example on a timeline

In this example scenario the following is assumed

- There is a file download service called “Music Clips” that represents a service delivering music related contents.
- The “Music Clips” service is delivered over MBMS using FLUTE as the transport method.
- Two music clips, “Music Video 1” and “Music Video 2”, are delivered in the file download service.
- “Music Video 1” is distributed from an undefined point in the past until 13:00 GMT on the 10<sup>th</sup> of July 2007.
- “Music Video 2” is distributed between 11:00 GMT and 12:00 GMT on the 10<sup>th</sup> of July 2007.
- Figure 30 above depicts the scenario on a timeline.

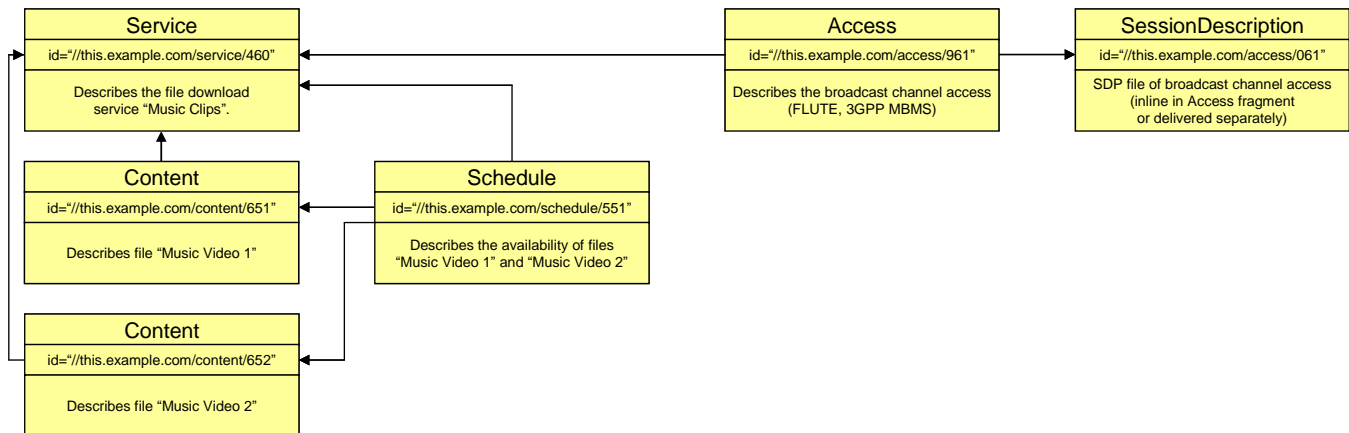


Figure 31: Example of Service Guide instantiation in case the delivered files are described in the Service Guide

The Service Guide is instantiated to support the scenario as follows:

- A single ‘Service’ fragment describes the “Music Clips” service. The ‘ServiceType’ attribute is set to 5 (“File download services”).
- For each of the files “Music Video 1” and “Music Video 2” there are an individual ‘Content’ fragment which together describe the files delivered within the service. These content fragments ‘inherit’ the ‘Access’ fragments associated with the ‘Service’ fragment as specified in section 5.8.4.1.
- ‘Schedule’ fragment with identifier “//this.example.com/schedule/551” schedules the content associated with service “Music Clips” and references the ‘Service’ fragment in addition to all of the ‘Content’ fragments. One instantiation of the ‘ContentReference’ element is used for each of the ‘Content’ fragments:
  - The first instantiation of ‘ContentReference’ has the ‘idRef’ attribute referencing ‘Content’ fragment “//this.example.com/content/651”. The ‘contentLocation’ attribute corresponds to the ‘Content-Location’ attribute identifying file “Music Video 1” in the FDT of the FLUTE session. ‘DistributionWindow.startTime’ is not instantiated (which corresponds to a time in the past) and

- 'DistributionWindow.endTime' is set to 3393061200 (which corresponds to 13:00 GMT on 10<sup>th</sup> of July 2007).
- The second instantiation of 'ContentReference' has the 'idRef' attribute referencing 'Content' fragment "//this.example.com/content/652". The 'contentLocation' attribute corresponds to the 'Content-Location' attribute identifying file "Music Video 2" in the FDT of the FLUTE session. 'DistributionWindow.startTime' is set to 3393054000 (which corresponds to 11:00 GMT on 10<sup>th</sup> of July 2007) and 'DistributionWindow.endTime' is set to 3393057600 (which corresponds to 12:00 GMT on 10<sup>th</sup> of July 2007).
  - 'Access' fragment with identifier "//this.example.com/access/961" describes the access to "Music Clips". Sub-element 'BroadcastServiceDelivery' is instantiated to declare that this access is over the broadcast channel. The attribute 'Access.BroadcastServiceDelivery.BDSType.type' is set to "1 – 3GPP MBMS" to denote that the delivery takes place over MBMS bearer and the 'Access.BroadcastServiceDelivery.FileDescription' element is omitted to signal that FLUTE is used as the transport method. The value of the 'ServiceClass' element of this 'Access' fragment is set to "urn:oma:bcast:oma\_bsc:fc:1.0".
  - 'SessionDescription' fragment with identifier "//this.example.com/access/061" is instantiated as SDP description and associated with 'Access' fragment "//this.example.com/access/961" and hence describes the way to access the service "Music Clips" over the broadcast channel. Alternatively, the 'SessionDescription' could be instantiated as an inline element within the 'Access' fragment.
  - Figure 31 above depicts the necessary Service Guide fragments and their relations.

## C.12 Scenarios of Unicast Service Guide requests

The access of the Service Guide over the Interaction Channel may be beneficial as depicted in section 5.5.3 in several commercial deployment contexts.

This section describes typical use cases for requesting the Service Guide over the Interaction Channel. For each use case, it is exhibited typical requests that should be supported by servers and terminals.

### C.12.1 LIVE TV services

#### C.12.1.1 Service plan and provisioning information retrieval

In order that the terminal can discover and access services as fast as possible, the first step of the Service Guide access over the Interaction Channel could be to discover the services, their access, and the related provisioning information ('PurchaseItem', and related 'PurchaseData' fragments).

##### C.12.1.1.1 Related requests and responses

Terminals may use the following conjunction of key-value-pairs:

```
type=sgdu&serviceType=1&all=true&function=serviceAccess&function=servicePurchase
```

The terminal is returned the list of basic TV Service fragments, and Access fragments that reference directly to the services. The response also contains the Schedule fragments that reference to those particular services, but do not reference to other fragments, with their related Access fragments and the PurchaseItem fragments that reference directly to the services, the PurchaseData that reference to those PurchaseItem fragments.

The PurchaseChannel fragment(s) that is referenced to by the PurchaseData fragment obtained as the result of the above request, may be fetched thanks to the following request:

```
type=sgdu&fragmentID="PurchaseChannelX"
```

where "PurchaseChannelX" is the identifier of the requested PurchaseChannel fragment.

If a terminal wants also to retrieve the list of PreviewData fragments (and the related Access fragments) that are referenced to by the Service fragments, then, it may send the following conjunction of key-value pairs:

```
type=sgdu&serviceType=1&all=true&function=serviceAccess&function=servicePurchase&function=servicePreview
```

### C.12.1.1.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value-pairs for this particular use case:

```
type=sgdu&serviceType=1&all=true&fragmentType=1&fragmentType=4&fragmentType=5&fragmentType=6
```

Thanks to this request, the terminal is returned the list of Service, Access, PurchaseItem, PurchaseData fragments that relate to Basic TV services.

Note that some of the returned Access fragments may not relate directly to the access to services (e.g. may relate to access to interactivity). The same applies to Purchase Item, Purchase Data fragments that may reference e.g. Contents or Schedule fragments.

Further, the PurchaseChannel fragment(s) that is referenced to by the PurchaseData fragment(s) obtained as the result of the above request, may be fetched thanks to the following request:

```
type=sgdu&fragmentID="PurchaseChannelX"
```

where "PurchaseChannelX" is the identifier of the requested PurchaseChannel fragment.

In deployments contexts where the Session Description fragments are not directly embedded in the Access fragments, the BCAST 1.0 terminal may use the following conjunction of key-value-pairs to retrieve these Session Description fragments:

```
type=sgdu&fragmentEncoding=1&serviceType=1&all=true
```

### C.12.1.2 Electronic Program Guide information

It might be beneficial that after having fetched the service plan and provisioning information descriptions, the terminal fetches content, schedule, interactivity descriptions. Access and provisioning information that relate to these fragments should be acquired, as well.

#### C.12.1.2.1 Related requests and responses

Terminals that want to retrieve fragments that describe information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with a given time interval, may use the following conjunction of key-value pairs:

```
type=sgdu&startTime=X&endTime=Y&serviceType=1&globalContentIDAll=true&all=true&function=contentAccess&function=contentInteractivity&function=contentPurchase
```

where "X" and "Y" are respectively the 'startTime' and the 'endTime' values of the target time interval.

The terminal is returned the list of Content, Schedule, Interactivity Data fragments that match the requested time interval. It includes fragments describing access that reference these Schedule fragments and the access and the schedule information that relate to these Interactivity Media Documents. It also includes provisioning information (PurchaseItems, PurchaseData fragments) that reference directly to the Content (resp. the Schedule fragments).

This request may be confined by the <key> "globalServiceID" to retrieve fragments that relate to a given time interval and a given service.

When a terminal aims at having EPG information that relate to a relative period of time, it may use the following conjunction of key-value pairs:

```
type=sgdu&relativeStartTime=X&relativeEndTime=Y&serviceType=1&globalContentIDAll=true&all=true&function=contentAccess&function=contentInteractivity&function=contentPurchase
```

where "X" and "Y" are respectively the beginning and the end of the target relative time interval.

In the following example, we illustrate the benefit of such request:

A terminal wants to maintain near future EPG information, e.g 1 hour time:

It may send the following request:

```
type=sgdu&relativeEndTime=3600&serviceType=1&globalContentIDAll=true&all=true&function=contentAccess
&function=contentInteractivity&function=contentPurchase
```

The server may signal in the response an "expirationTime" value, so that the terminal will refresh its first request e.g. 1 hour later. Upon these conditions the terminal has always at least EPG information for the next one hour.

Note that for all these requests, the terminal may need also to retrieve the PreviewData fragments (and related Access fragments) that are referenced to by the Content (or Schedule fragments) that match the target period of time. In that case, the terminal may use in addition the key-value "function=contentPreview".

### C.12.1.2.2 Related requests initiated by BCAST 1.0 terminals

Terminals that want to retrieve fragments that describe information related to content or interactivity that can be distributed, consumed, or activated during a time interval that is not disjoint with a given time interval, may use the following conjunction of key-value pairs:

```
type=sgdu&tgc-start=X&tgc-end=Y
```

where "X" and "Y" are respectively the 'startTime' and the 'endTime' values of the target time interval.

Thanks to this request the terminal is returned the list of Content, Schedule, and InteractivityData fragments that match the requested time interval. It also includes Access fragments that reference these Schedule fragments and the Schedule and Access information that relate to the delivery of the Interactivity Media Documents. It includes provisioning information (PurchaseItem, PurchaseData fragments) that reference directly to the Content (resp. the Schedule fragments), Preview Data that are referenced to by the Content fragments (resp. the Schedule fragments) and the Access information that are referenced to by the Preview.

This request may be confined by the <key> "srvc" to retrieve fragments that relate to a given time interval and a given service.

Note that the time interval that is declared in this kind of request is an absolute time.

Terminals that aim at having near future EPG information may use the following conjunction of key-value pairs:

```
type=sgdu&fragmentType=2&fragmentType=3&fragmentType=9&serviceType=1&all=true
```

In this case, it is server implementation specific to decide the time interval of the returned Content, Schedule and Interactive fragments have to be based on.

This request may be confined by the <key> "globalServiceID" so that the list of returned Content, Schedule, Interactive fragments relate to the declared Service.

### C.12.1.3 Preview information retrieval

#### C.12.1.3.1 Related requests and responses

Terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments, if not received in a previous step (C.13.1.1.1, resp. C.13.1.2.1):

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers obtained from the previous step (C.13.1.1.1, resp. C.13.1.2.1)

#### C.12.1.3.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers obtained from the previous step (C.13.1.1.2, resp. C.13.1.2.2)

Note that BCAST 1.0 terminals may use the following conjunction of key-value pairs to retrieve the entire set of PreviewData fragments:

```
type=sgdu&fragmentType=8&serviceType=1&all=true
```

Note however that some of the returned PreviewData fragments within the entire set may not be of immediate interest.

## C.12.2 ON DEMAND SERVICES

### C.12.2.1 Audio Video On Demand services retrieval

Terminals that want to retrieve fragments that describe information related to "Audio Video On Demand" may benefit from discovering the set of Services that provide such contents, and the provisioning information ("PurchaseItem" and related "PurchaseData" fragments). Note that for "Audio Video On Demand", the Access fragments are expected to be linked to the Content fragments through the related Schedule fragments. It might be beneficial to retrieve those Access fragments while requesting for the Content (and Schedule fragments).

#### C.12.2.1.1 Related requests and responses

Terminals may use the following conjunction of key-value-pairs:

```
type=sgdu&serviceType=11&all=true&function=servicePurchase
```

The terminal is returned the list of Service fragments of serviceType "Audio Video On Demand", the PurchaseItem fragments that reference directly to the Service fragments, and the PurchaseData fragments that reference to those PurchaseItem fragments.

Further, the PurchaseChannel fragment(s) that is referenced to by the PurchaseData fragment(s) obtained previously may be fetched thanks to the following request:

```
type=sgdu&fragmentID="PurchaseChannelX"
```

where "PurchaseChannelX" is the identifier of the requested PurchaseChannel fragment.

If a terminal wants also to retrieve the list of PreviewData fragments (and the related Access fragments) that are referenced to by the Service fragments, it may send the following conjunction of key-value-pairs:

```
type=sgdu&serviceType=11&all=true&function=servicePurchase&function=servicePreview
```

#### C.12.2.1.2 Related requests initiated by BCAST 1.0 terminals

There's no straightforward way for BCAST 1.0 terminals to request for "Audio Video On Demand" Service fragments.

### C.12.2.2 On demand contents retrieval

In the context of content on demand searching, it is expected that the terminals aim at knowing in detail the list of Content fragments that are available and the related access and provisioning information.

#### C.12.2.2.1 Related requests and responses

Terminals may use the following conjunction of key-value pairs:

```
type=sgdu&serviceType=11&globalContentIDAll=true&all=true&function=contentAccess&
function=contentInteractivity&function=contentPurchase&genre=X
```

where "X" is a particular genre.



The terminal is returned the list of Content, Schedule, InteractivityData fragments that relate to Audio Video On Demand that match the <key> "genre" equals to "X". It includes fragments describing access that reference these Schedule fragments and the access and the schedule information that relate to these Interactivity Media Documents. It also includes provisioning information (PurchaseItem, PurchaseData fragments) that reference directly to the Content or Schedule fragments.

This request may be confined by the <key> "globalServiceID" to retrieve fragments that relate to a given service.

Note that for all these requests, the terminal may need also to retrieve the PreviewData fragments (and related Access fragments) that are referenced to by the target "Audio Video On Demand" Content (or Schedule) fragments. In that case, the terminal may use in addition the key-value pair "function=contentPreview".

### C.12.2.2.2 Related requests initiated by BCAST 1.0 terminals

There's no straightforward way for a BCAST 1.0 terminals to discover the list of Audio Video On Demand contents.

## C.12.2.3 Preview information retrieval

### C.12.2.3.1 Related requests and responses

Terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments, if not received in a previous step (C.13.2.1.1, resp. C.13.2.2.1):

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers obtained from the previous step (C.13.2.1.1, resp. C.13.2.2.1)

### C.12.2.3.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments:

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers retrieved in a previous step

## C.12.3 CACHE CAST DOWNLOADING

### C.12.3.1 Service plan and provisioning information retrieval

The first step of the Service Guide access over the Interaction Channel could be to discover the services, and the related provisioning information ('Purchase Item', related 'Purchase Data'). Note that for Carousel file downloading, the Access fragments are expected to be linked to the Content fragments through the related Schedule fragments. It might be beneficial to retrieve those Access fragments while requesting for the Content (and Schedule fragments), themselves.

### C.12.3.1.1 Related requests and responses

Terminals may use the following conjunction of key-value-pairs.

```
type=sgdu&serviceType=4&all=true&function=servicePurchase
```

The terminal is returned the list of Service fragments of serviceType "4" (Cachecast), the PurchaseItem fragments that reference directly to the Service fragments, and the PurchaseData fragments that reference to those PurchaseItem fragments.

Further, the PurchaseChannel fragment(s) that is referenced to by the PurchaseData fragment(s) obtained previously may be fetched thanks to the following request:

```
type=sgdu&fragmentID="PurchaseChannelX"
```

where "PurchaseChannelX" is the identifier of the requested PurchaseChannel fragment

If a terminal wants also to retrieve the list of PreviewData fragments (and the related Access fragments) that are referenced to by the Service fragments, it may send the following conjunction of key-value-pairs:

```
type=sgdu&serviceType=4&all=true&function=servicePurchase&function=servicePreview
```

### C.12.3.1.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value-pairs for this particular use case.

```
type=sgdu&fragmentType=1&fragmentType=5&fragmentType=6&serviceType=4&all=true
```

Thanks to this request, the terminal is returned the list of related Service fragments, PurchaseItem and PurchaseData fragments. Note that some of the PurchaseItem, PurchaseData fragments may reference e.g. Content or Schedule fragments and not directly the Service fragments themselves.

Further, the PurchaseChannel fragment(s) that is referenced to by the PurchaseData fragment(s) obtained previously may be fetched thanks to the following request:

```
type=sgdu&fragmentID="PurchaseChannelX"
```

where "PurchaseChannelX" is the identifier of the requested PurchaseChannel fragment.

### C.12.3.2 Cachecast content retrieval

In the context of Cachecast Content searching, it is expected that the terminals aim at knowing in detail the list of Content fragments that are available and the related access and provisioning information.

#### C.12.3.2.1 Related requests and responses

Terminals may use the following conjunction of key-value pairs:

```
type=sgdu&serviceType=4&globalContentIDAll=true&all=true&function=contentAccess&
function=contentInteractivity&function=contentPurchase&genre="X"
```

where "X" is a particular genre.

The terminal is returned the list of Content, Schedule, InteractivityData fragments that relate to Cachecast content and that match the <key> "genre" equals to "X". It includes fragments describing Access fragments that reference these Schedule fragments and the access and the schedule information that relate to these InteractivityData fragments. It also includes provisioning information (PurchaseItem, PurchaseData fragments) that reference directly to the Content (resp. the Schedule) fragments.

This request may be confined by the <key> "globalServiceID" to retrieve fragments that relate to a given service.

Note that for all these requests, the terminal may need also to retrieve the PreviewData fragments (and related Access fragments) that are referenced to by the target "Cachecast" Content (or Schedule) fragments. In that case, the terminal may use in addition the key-value pair "function=contentPreview".

#### C.12.3.2.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value pairs to discover the presence of Cachecast contents:

```
type=sgdu&fragmentType=2&fragmentType=3&fragmentType=4&fragmentType=9&serviceType=4&genre="X"
&all=true
```

Note that the list of returned fragments are scoped by the <key> "genre" which value equals to "X"

In deployments contexts where the Session Description fragments are not directly embedded in the Access fragments, then the BCAST 1.0 terminal may use the following conjunction of key-value-pairs to retrieve the Session Description fragments:

```
type=sgdu&fragmentID="X"(&fragmentID="Y",...)
```

to retrieve the list of session description fragments which identifiers are "X" (resp. "Y").

### C.12.3.3 Preview information retrieval

#### C.12.3.3.1 Related requests and responses

Terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments, if not received in a previous step (C.13.3.1.1, resp. C.13.3.2.1):

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers retrieved in the first step (C.13.3.1.1, resp. C.13.3.2.1).

#### C.12.3.3.2 Related requests initiated by BCAST 1.0 terminals

BCAST 1.0 terminals may use the following conjunction of key-value pairs to retrieve a set of PreviewData fragments:

```
type=sgdu&fragmentID="X"(&fragmentID="Y", ...)
```

where "X" and "Y" are values of referenced PreviewData fragments identifiers retrieved in a previous step (C.13.3.1.2, resp. C.13.3.2.2)

Note that BCAST 1.0 terminals may use the following conjunction of key-value pairs to retrieve the entire set of PreviewData fragments:

```
type=sgdu&fragmentType=8&serviceType=4&all=true
```

Note however that some of the returned PreviewData fragments within the entire set may not be of immediate interest.

### C.12.4 General practice applicable to all scenarios

As explained in section 5.4.3.1, it is beneficial that the terminal instantiates the "lastResponseVersion" key in the request provided that this information was returned by the server in a response to a previous identical request.

It is also beneficial that the server adds the ResponseValidity element in the response.

To gain bandwidth, the response should be compressed using the GZIP algorithm.

## Appendix D. Example of SGDD and SGDU

The following diagram provides an illustrative example of the relationship between SGDD and SGDU, as well as the containers in which they reside – SG Announcement Channel and SG Delivery Channels, respectively. In this example, it is assumed that the grouping of SG fragments is by time in the delivery over hourly, daily and weekly SG delivery channels.

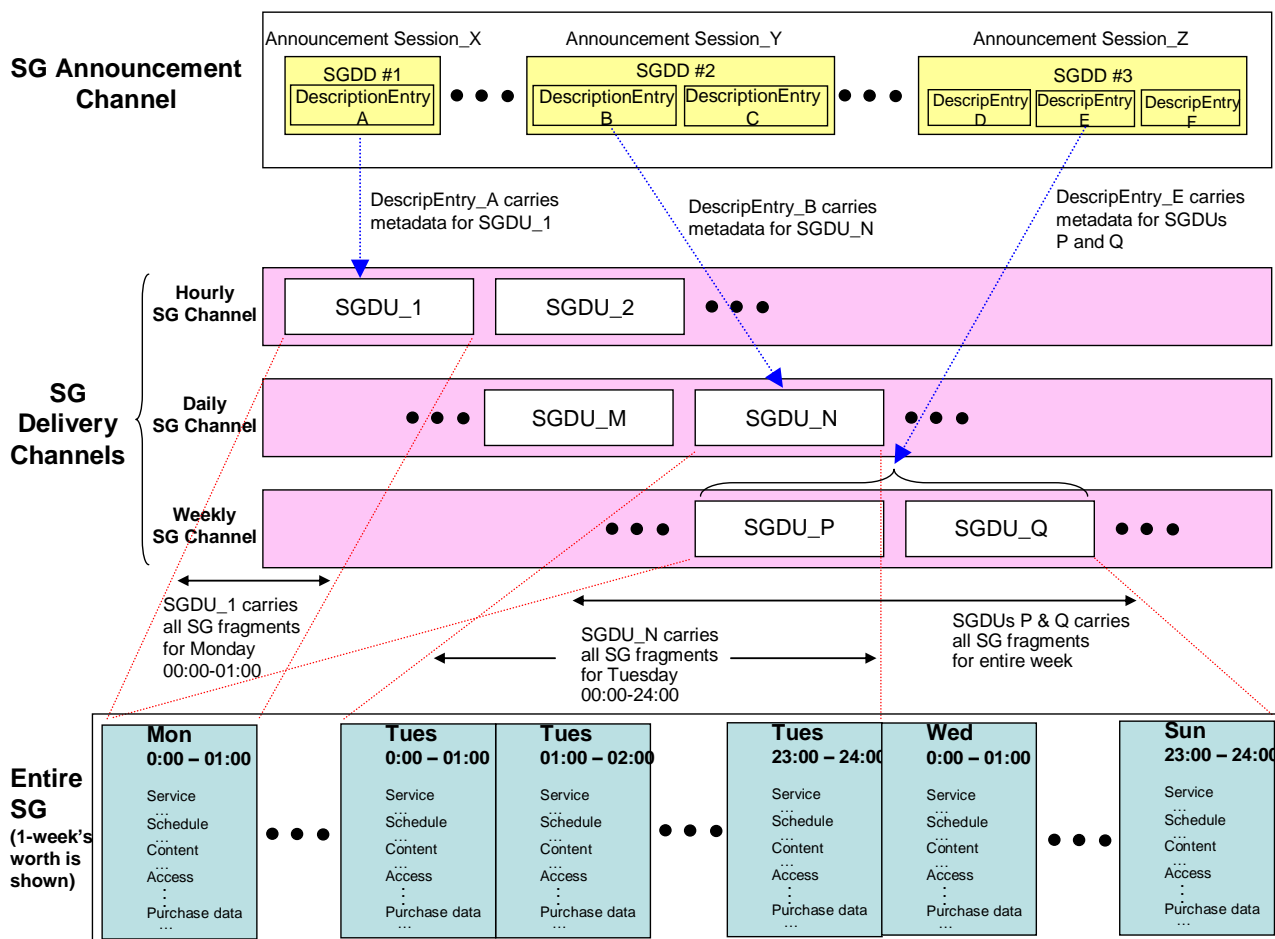


Figure 32: Service Guide Delivery Descriptor role in Service Guide delivery showing SG fragments grouped by time

## Appendix E. Service Class Identifier Handling (Normative)

### E.1 Definition of service class values

#### E.1.1 Service Class Definition

A service class identifier is a unique name used to identify a particular service class. The service class is a combination of many properties, such as used media formats, file types, handling of media, interaction with user, etc, which describes how a particular service works. A service class SHALL be documented in a specification, and a unique service class identifier SHALL be assigned to it, following the syntax and procedure in E.1.2.. Individual service providers may not have a formal specification for a service, however the particular capabilities of a specific implementation of the service is recommend to be provided with a service class identifier.

The service class identifier is provided with the 'Access' fragment and allows a terminal to match services and applications.

#### E.1.2 Service Class Naming and Registration

The Open Mobile Naming Authority [OMNA] maintains a registry of service classes. The service class names in this registry are structured as a URN to assure uniqueness. The model follows the following grammar:

```
{ClassName} ::= { {OMName} | {ExtName} }
{OMName}    ::= urn:oma:bcast:oma_bsc:{ClassLabel}:{Version}
{ExtName}   ::= urn:oma:bcast:ext_bsc_{Org}:{ClassLabel}:{Version}
```

Where:

- {ClassLabel} is a string made up of lowercase alpha characters
- {Version} is a string consisting of numbers and periods (e.g. '1.2')
- {Org} is a string that conveys the name of the external organization

Service classes defined in an OMA enabler SHALL be registered with [OMNA], providing a description of the service, a reference to the specification where it is documented and a service class identifier proposal following the above syntax for {OMName}.

External organizations defining service classes are RECOMMENDED to register a URN prefix for their service class identifiers with [OMNA] and publish their service classes in their own registry. Uniqueness of service class identifiers can then be managed as the prefix will be unique to the registering organization and the {ClassLabel}:{Version} need only be unique within that organization's local registry.

To assist with communication of service classes defined by external organizations, [OMNA] will provide links to registration information available from these groups, as appropriate. Organizations seeking a prefix assignment are advised to check with [OMNA] for information.

### E.2 Service Classes defined in OMA BCAST

This section defines the service classes for the OMA BCAST enabler.

#### E.2.1 urn:oma:bcast:oma\_bsc:sg:1.0

Services of this service class deliver metadata fragments using a broadcast channel or the interactive channel. The discovery of Service Guide is defined in section 6.

#### E.2.2 urn:oma:bcast:oma\_bsc:fc:1.0

Services of this service class offer a file carousel service, using the File Delivery function of OMA BCAST. Files are transmitted and locally stored in a file cache. Files are uniquely identified by the URI. Files in the file carousel are periodically re-transmitted. It may happen that already received files are replaced by a new file in a later repetition. An example of a file carousel is a teletext service.

### **E.2.3 urn:oma:bcast:oma\_bsc:tp:1.0**

Services of this service class offer a Terminal Provisioning service either using the broadcast channel over TP-5 or interaction channel over TP-7. The management objects delivered over sdo.oma.tp SHALL be of MIME type "application/vnd.syncml.dm+wbxml". [BCAST12-Services] defines Terminal Provisioning in section 5.2.

### **E.2.4 urn:oma:bcast:oma\_bsc:nt:1.0**

Services of this service class deliver Notification Messages over NT-5 or NT-6. Both over the broadcast channel as well as over the interactive channel, the delivered messages SHALL follow the format as specified in section 5.14 of [BCAST12-Services]. The discovery and signaling of Notification Messages SHALL follow the specification in section 5.14.1 of [BCAST12-Services].

### **E.2.5 urn:oma:bcast:oma\_bsc:csg:1.0**

Services of this service class deliver a complementary Service Guide. The discovery of Service Guide is defined in section 6.

### **E.2.6 urn:oma:bcast:oma\_bsc:st:1.0**

Services of this service class offer a streaming service, using the Stream Delivery function of OMA BCAST. Streams are transmitted as specified in section 6 of [BCAST12-Distribution]. Examples of this service are live TV and live Radio, realized as streaming audiovisual or audio-only services.

### **E.2.7 urn:oma:bcast:oma\_bsc:rifc:1.0**

Services of this class offer a file carousel service for transmission of Registration Layer and Rights Management Layer objects and messages. RI Services are specified in section 12 of [DRM20-Broadcast-Extensions].

### **E.2.8 urn:oma:bcast:oma\_bsc:fd:1.1**

Services of this service class offer a file download service over the Interaction channel, using the File Delivery function of OMA BCAST. Files are transmitted and locally stored in a file cache. Files are uniquely identified by the URI. An example of such service is an AV content downloading.

### **E.2.9 urn:oma:bcast:oma\_bsc:sp:1.1**

Services of this service class offer a Smartcard Provisioning service using the broadcast channel over FD-5. Files are transmitted and locally stored in a file cache and then transmitted to the Smartcard.

## Appendix F. Global Status Codes

The Global Status Codes specified in [BCAST12-Services] lists all the possible status codes for error case, and their applicability to each transaction. The codes are to be used for Response in SG-4 and NT-1, NT-3, NT-4 response messages in this specification. The codes may also be used in other response messages in other BCAST technical specifications.

The Table 11 below shows example values from the Global Status Codes specified in [BCAST12-Services] for the transaction messages that require the use of Global Status Codes. The values shown below are for informative purposes and the full range of values of the Global Status Codes are applicable to all messages if deemed required.

TS-BCAST_ServiceGuide	5.6.2.2.2 Response Message for content delivery	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	5.6.2.3 Response Message for service protection information delivery	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	5.6.2.4 Response Message for service provisioning information delivery	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	7.5.2.2 Response Message	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	7.5.3.2 Response Message	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023

**Table 11: Cross Reference Table (Informative)**

## Appendix G. MIME media types

### G.1 Media-Type Registration Request for application/vnd.oma.bcast.sgdu

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.sgdu
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

#### Security considerations:

BCAST Service Guide Delivery Unit data are passive, meaning they do not contain no executable or active content which may represent a security threat. The format BCAST Service Guide Delivery Units does not contain include confidential fields which are confidential. However, the information present in this media format is used to configure the receiving application. Thus, the usage of the format may be vulnerable to attacks modifying or spoofing the content of this format. Depending on the system architecture, it is recommended to use source authentication and integrity protection.

#### Interoperability considerations:

This content type carries service guide fragments within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

#### Published specification:

OMA BCAST 1.0 Enabler Specification – Service Guide for Mobile Broadcast Services, especially section 5.4.1.3. Available from <http://www.openmobilealliance.org>

#### Applications, which use this media type:

OMA BCAST Service Guide

#### Additional information:

Magic number(s):	none
File extension(s):	none
Macintosh File Type Code(s):	none

Intended usage: Limited use.

Only for usage with BCAST Service Guide Delivery Unit, which meet the semantics given in the mentioned specification.



Person & email address to contact for further information:

Uwe Rauschenbach

[Uwe.Rauschenbach@nsn.com](mailto:Uwe.Rauschenbach@nsn.com)

Author/Change controller:

OMNA – Open Mobile Naming Authority, [OMA-OMNA@mail.openmobilealliance.org](mailto:OMA-OMNA@mail.openmobilealliance.org)

## G.2 Media-Type Registration Request for application/vnd.oma.bcast.sgdd+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.sgdd+xml
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:

BCAST Service Guide Delivery Descriptor data are passive, meaning they do not contain no executable or active content which may represent a security threat. Also, the format BCAST Service Guide Delivery Descriptor data does not contain include confidential fields which are confidential. However, the information present in this media format is used to configure the receiving application. Thus, the usage of the format may be vulnerable to attacks modifying or spoofing the content of this format. Depending on the system architecture, it is recommended to use source authentication and integrity protection.

Interoperability considerations:

This content type carries service guide delivery information within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – Service Guide for Mobile Broadcast Services, especially section 5.4.1.5. Available from <http://www.openmobilealliance.org>

Applications, which use this media type:

OMA BCAST Service Guide

Additional information:

Magic number(s): none

File extension(s): none

Macintosh File Type Code(s): none

Person & email address to contact for further information:

Uwe Rauschenbach

[Uwe.Rauschenbach@nsn.com](mailto:Uwe.Rauschenbach@nsn.com)

Intended usage: Limited use.

Only for usage with BCAST Service Guide Delivery Descriptor, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, [OMA-OMNA@mail.openmobilealliance.org](mailto:OMA-OMNA@mail.openmobilealliance.org)

## Appendix H. Identification of unique elements (Normative)

In order to permit unique identification of items in the Service Guide, item URI have to be constructed in a way that allows such unique identification. This appendix provides a method for constructing such URIs. Implementations MAY support this method and the provisions of this appendix.

### H.1 Definition of BCAST Identification Authority

Identification Authorities are responsible for providing unique identifiers within their domain of control. Global uniqueness of items is then ensured by having the Identification Authority registered in a common registry.

The Open Mobile Naming Authority [OMNA] maintains such registry of Identification Authorities. The Identification Authority name in this registry is structured as a URN to ensure uniqueness. The model follows the following grammar:

```
{IdentAuthLabel} ::= urn:oma:bcast:iauth:{IAiD}
```

Where *{IAiD}* is a string assigned to a BCAST Identification Authority. It SHALL be made up of lowercase alphanumeric characters and the characters “-” and “.” where the latter two characters SHALL not hold the first and last positions in the string.

Organizations willing to act as Identification Authorities providing item identification in the BCAST Service Guide SHOULD register with [OMNA]. To assist in this, [OMNA] will provide links to registration information available as appropriate. Organizations seeking a prefix assignment are advised to check with [OMNA] for information. [OMNA] registered *{IAiD}* values are restricted to lowercase alphanumeric characters.

Note 1: for testing purposes, the pattern “x-\*”, where “\*” is a string of lowercase alphanumeric characters, is reserved for the *{IAiD}* in the OMNA registry. This pattern is not open for registration and MAY be used for private values.

Note 2: an alternative to registering an *{IAiD}* with [OMNA] is to use a domain name registered to the Identification Authority as part of the *{IAiD}*. In such case, the format of *{IAiD}* SHALL be “dns-\*” where “\*” is a valid domain name according to [RFC 1035] section 2.3.1, with the update defined in [RFC 1123] section 2.1. It should be noted, however, that using domain names can cause collision problems for items with a long lifetime when the owner of the domain name changes. This pattern is not open for registration.

### H.2 Identification of content with baseCID

Unique identification of one content is done by an Identification Authority, extending its own identification URN into a URI. The model follows the following grammar:

```
{BaseCID} ::= {IdentAuthLabel}:basecid:{BaseCIDExt}
```

Where :

- *{IdentAuthLabel}* is the URN of the Identification Authority as defined in section H.1.
- *{BaseCIDExt}* is an extension string that identifies the content within the scope of the Identification Authority.
- *{BaseCID}* provides a fully qualified identification of the content that is usable as a ‘baseCID’ in the ‘Service’ and ‘Content’ fragment.

The *{BaseCID}* URIs are communicated in the Service Guide and used in the construction of program\_CID and service\_CID. They SHALL comply with the syntax rules specified for program\_CID and service\_CID in Section 5.5.1 of [BCAST12-ServContProt]), in particular to the syntax of the ‘local-part’ of the ‘cid-urI’. This means:

- Escaped Encoding according to RFC 2396 MUST be applied to disallowed characters before the *{BaseCID}* URI is included in the Service Guide ‘baseCID’ attribute
  - in particular, the “.” character MUST be replaced by “%3A”.

- The *{BaseCIDExt}* part, which is under the responsibility of the Identification Authority, SHALL also comply with those restrictions.

## Appendix I. Service Guide Delivery in Hybrid Broadcast/Interactive Scenarios (INFORMATIVE)

A hybrid Mobile TV system is a mobile media delivery system where services are available over both a broadcast bearer (broadcast channel, such as DVB-H, MBMS, or BCMCS) and a 3G unicast bearer (interaction channel). This section describes possibilities and typical usage scenarios of such hybrid Mobile TV system, exhibiting one or more of the following characteristics.

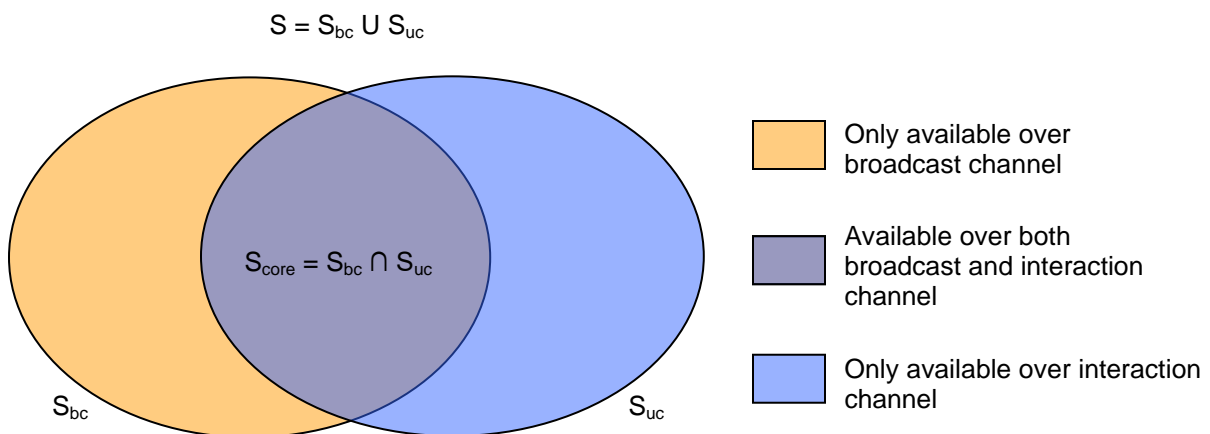
- a) The broadcast channel is unavailable in certain parts of the target area (e.g. rural areas or indoor areas), and the interaction channel is used to extend the coverage towards those areas. This way the interaction channel provides a fallback delivery method.
- b) The interaction channel is available in areas where also the broadcast channel is available. The interaction channel offers certain services that are not available over the broadcast channel (e.g. on-demand video, some parts of the service during scheduled bearer switching).

In case a), the media settings (e.g. bit rates) of services available over both the interaction channel and the broadcast channel are typically the same. In case b), a service that is available over the broadcast channel may be alternatively available over the interaction channel with different media settings (e.g. different quality).

The following sections describe aspects related to delivery and management of Service Guide in a hybrid Mobile TV system.

### I.1 File and Stream Distribution Services

Overall, a set  $S$  of file and stream distribution services is available in the hybrid Mobile TV system as shown in Figure 33. Services available over the broadcast channel are denoted as  $S_{bc}$  and services available over the interaction channel are denoted as  $S_{uc}$ . Some services in  $S_{bc}$  are only available over the broadcast channel. Some services in  $S_{uc}$  are only available over the interaction channel. The set of services available over both broadcast channel and interaction channel is denoted as  $S_{core}$  which is the intersection of the sets  $S_{bc}$  and  $S_{uc}$ . The relationship of the sets  $S_{bc}$ ,  $S_{uc}$  and  $S_{core}$  is illustrated below.



**Figure 33: General Services set diagram**

One realization of the general set diagram in Figure 33 is presented in Figure 34. It shows an example where all services are available both over the broadcast channel and over the interaction channel. This realization would typically correspond to the case where the interaction channel is used as a fallback delivery method in order to extend the coverage area.

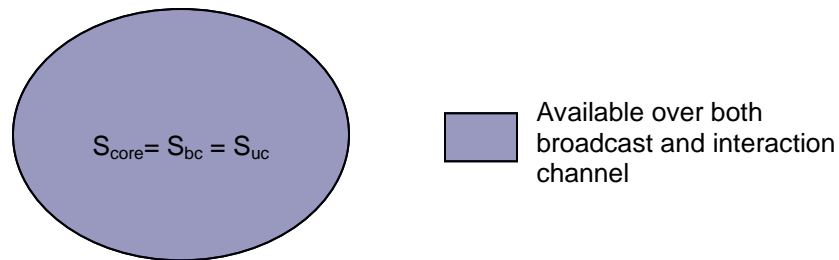


Figure 34: Typical services set diagram when the interaction channel is used as a fallback delivery method.

## I.2 Service Guide

The Service Guide, the set  $SG$  of fragments, consists of a set  $SG_{bc}$  of fragments that are available over the broadcast channel, and a set of fragments  $SG_{uc}$  that are available over the interaction channel. Some fragments in  $SG_{bc}$  are only available over the broadcast channel. Some fragments in  $SG_{uc}$  are only available over the interaction channel. The intersection of  $SG_{bc}$  and  $SG_{uc}$  is denoted as  $SG_{core}$  and describe the set of Service Guide fragment available over both broadcast channel and interaction channel. The relationship of those sets is illustrated in Figure 35.

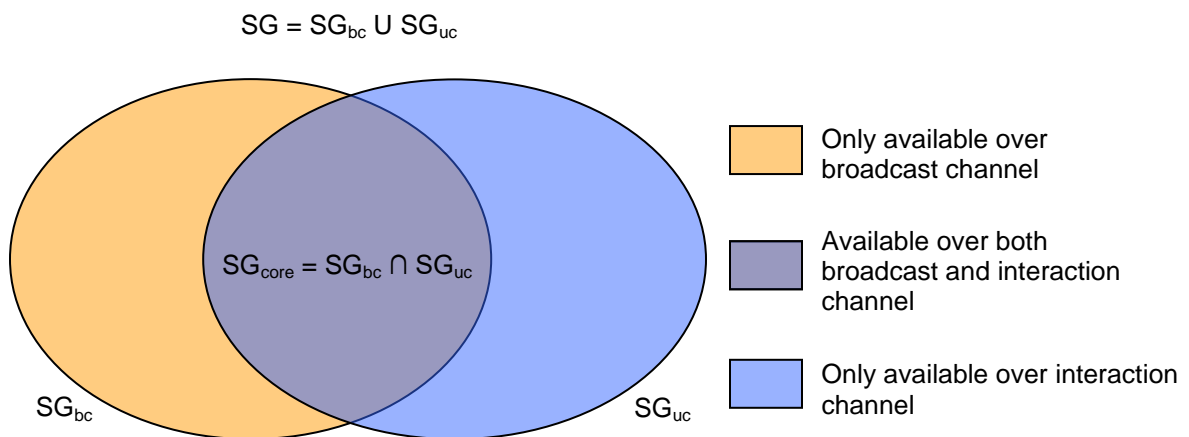
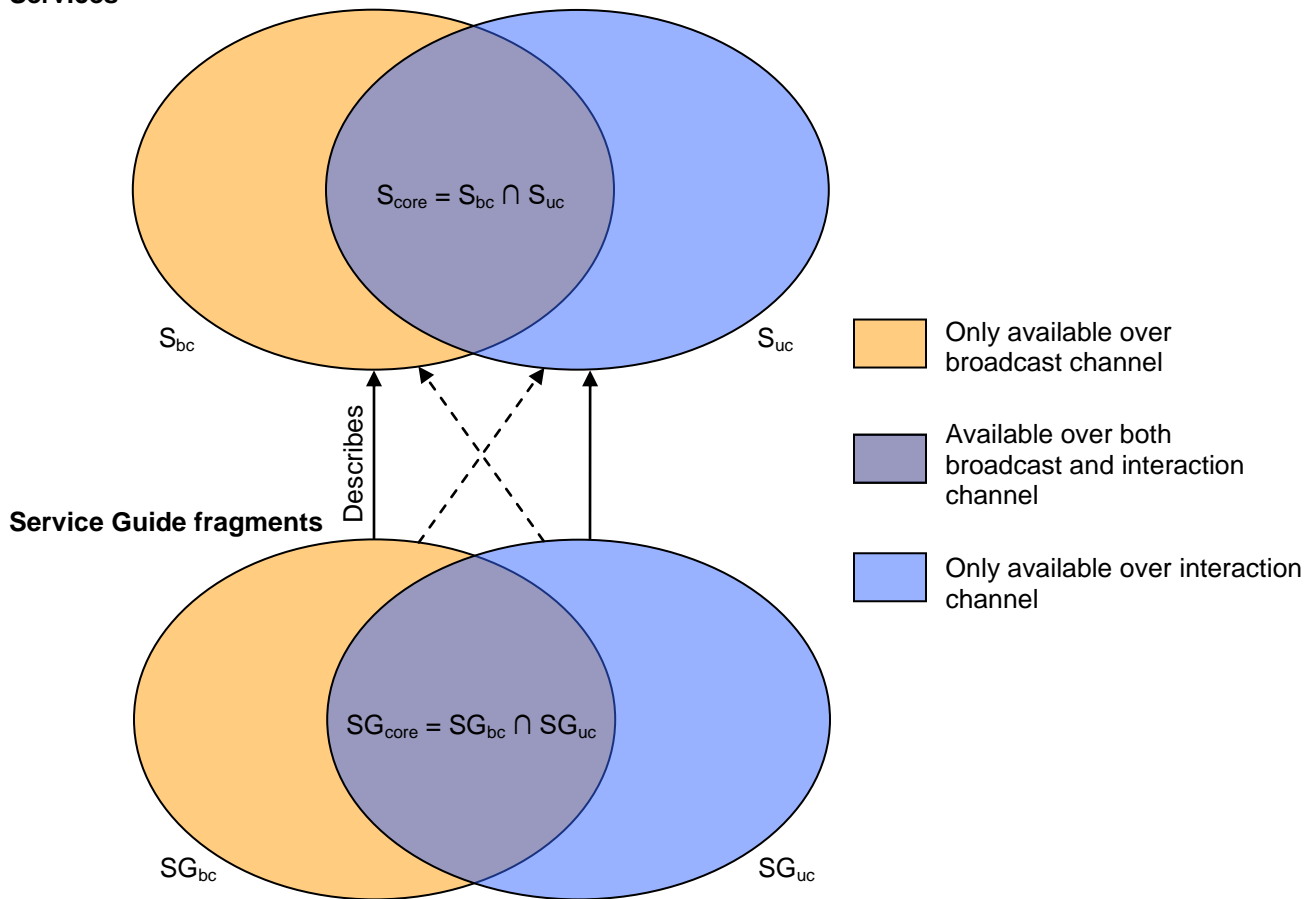


Figure 35: General Service Guide set diagram

If both the broadcast channel and the interaction channel are available, then it is up to the terminal to decide over which of them to receive  $SG_{core}$ . It may decide to retrieve  $SG_{core}$  over the broadcast channel, but it may alternatively decide to receive it over the interaction channel if it expects that to be beneficial in terms of resource usage or time. Fragments in  $SG_{bc}$  which are only available over the broadcast channel needs to be retrieved over the broadcast channel. Fragments in  $SG_{uc}$  which are only available over the interaction channel needs to be requested over the interaction channel. If only the broadcast channel is available, then the terminal retrieves  $SG_{bc}$  and if only the interaction channel is available, then the terminal retrieves  $SG_{uc}$ . The relationship between the service set diagram and Service Guide set diagram is shown in Figure 36.

Services



**Figure 36: The relation between the service set diagram and Service Guide set diagram**

If the set  $SG_{bc}$  is not empty (Service Guide fragments are available over the broadcast channel), then  $SG_{bc}$  typically contains at least all fragments that correspond to the services in  $S_{bc}$ . If the set  $SG_{uc}$  is not empty (Service Guide fragments are available over the interaction channel), then  $SG_{uc}$  typically contains at least all fragments that correspond to the services in  $S_{uc}$ . Additionally,  $SG_{bc}$  may contain fragments that correspond to certain services only available over the interaction channel (e.g. ‘Service’ fragment describing a VoD service, and possibly the corresponding ‘Access’ fragment that describes the interaction channel access). Correspondingly,  $SG_{uc}$  may contain some fragments that correspond to certain services only available over the broadcast channel (e.g. announcement of services that are only available over the broadcast channel).

Figure 37 illustrates the specific and typical case where the whole Service Guide is available over the interaction channel and parts of the Service Guide are alternatively available over the broadcast channel. This realization would typically correspond to the case where the interaction channel offers certain services that are not available over the broadcast channel (e.g., on-demand services). In this example, the fragments describing these additional services are then only available over the interaction channel.

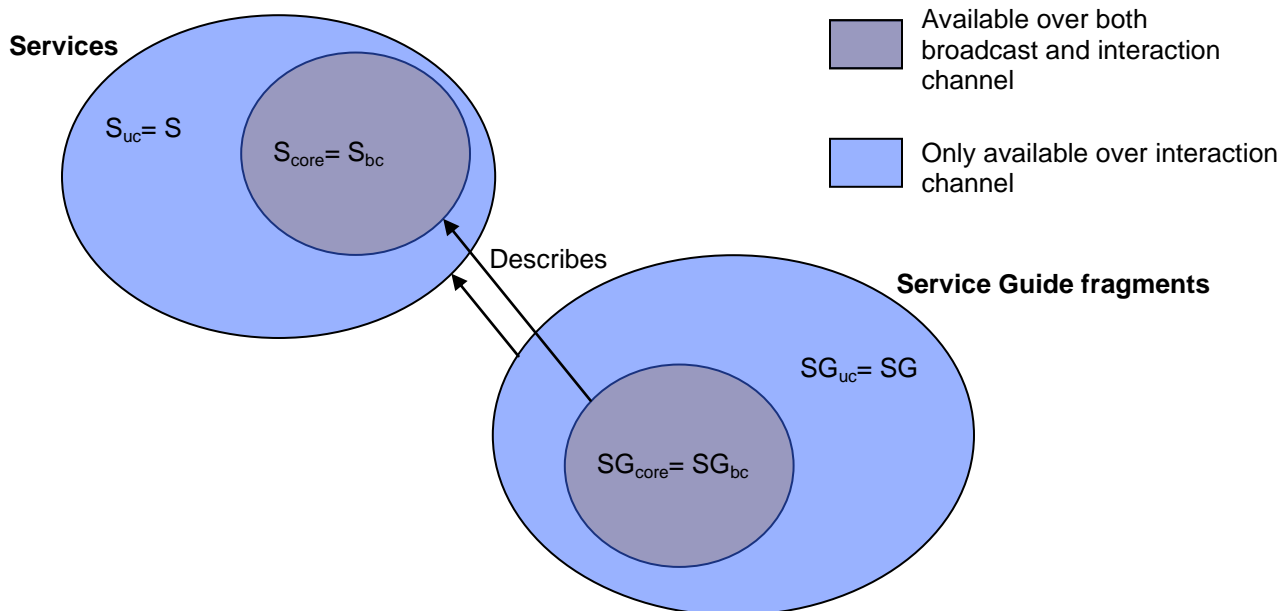


Figure 37: Typical set diagram when the interaction channel extends the service offering.

## I.2.1 Hybrid Broadcast/Unicast Service Guide Discovery

The entry point for  $SG_{bc}$  is determined according to section 6.1.

If the broadcast channel is available, then an entry point of  $SG_{uc}$ , along with its relationship with  $SG_{bc}$ , can be retrieved through ‘UnicastServerURL’ in SGDDs delivered over the broadcast channel. This URL allows for specific requests for fragments declared in the respective SGDD, or for unspecific requests of  $SG_{uc}$ . Alternatively, e.g., in case the broadcast channel is not available, the entry point for  $SG_{uc}$  is determined according to section 6.2.

## I.2.2 Hybrid Broadcast/Unicast Service Guide Update and Management

See section 5.5.3 for a description of hybrid broadcast/unicast Service Guide update and management.

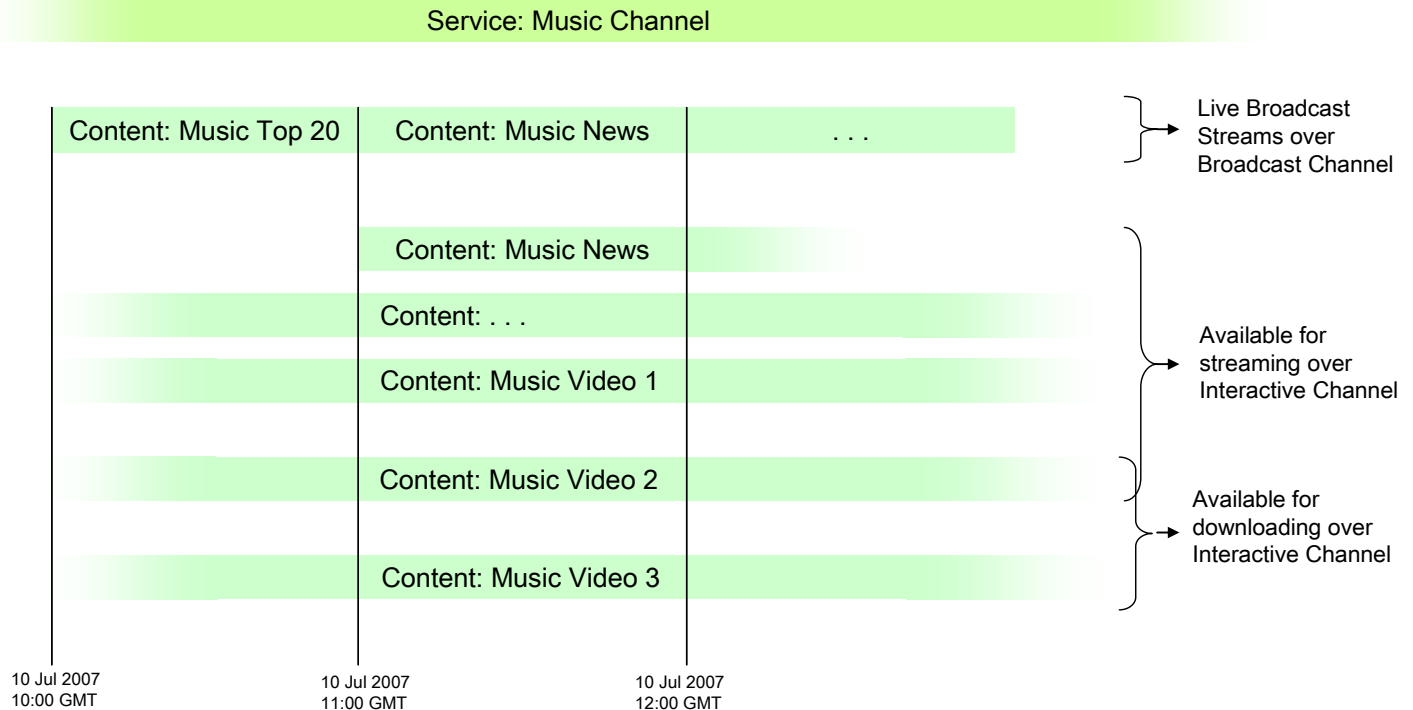
## I.3 Hybrid Broadcast/Unicast Service Guide Instantiation

This section describes two scenarios of how a Service Guide can be instantiated in a hybrid broadcast/unicast environment. Section I.3.1 describes a scenario where some part of the content is only available over the broadcast channel, some of the broadcasted content are also alternatively available over the interaction channel, and finally some part consisting of on-demand content which is naturally only available over the interaction channel. Section I.3.2 describes a similar scenario but with the difference that all the broadcasted content is also alternatively available over the interaction channel, i.e., it is a typical scenario for when the interaction channel is used as a fallback delivery method. Section I.3.3 describes a scenario where a file download service available over broadcast channel is also alternatively available over the interaction channel.

### I.3.1 Scenario of Service Guide with both Broadcast and Unicast Services

A common use of Service Guide is to enable the discovery of both broadcast and unicast services and content within a single place. The application scenario in this section explains how the Service Guide can be instantiated to support such use. In this application scenario, broadcast services are provided over the Broadcast Channel and are live, unidirectional, multicast streaming services. The unicast services are provided over the Interaction Channel and are streaming services available per request, on-demand.





**Figure 38: Service example on a timeline**

In this example scenario the following is assumed

- There is a service called “Music Channel” that represents a mobile TV / video service delivering music related contents. The users are able to access the “Music Channel” in two ways – either receiving the broadcast streams over Broadcast Channel or accessing the contents over Interaction Channel. These two ways are uniformly represented as a single service on the user interface.
- On the Broadcast Channel the “Music Channel” delivers content so that individual programs such as “Music Top 20” and “Music News” are sequentially scheduled. That means, as the “Music Top 20” ends, it is followed by the next program “Music News”.
- Some of the contents associated with service “Music Channel” are available over the Interaction Channel so that the user can select those for streaming. In the example, “Music News”, “Music Video 1” and “Music Video 2” are such contents. “Music Video 1” and “Music Video 2” do not have a defined availability time meaning these streams have been available since undefined point of time and will be available until undefined point of time for streaming. “Music News”, on the other hand, is scheduled to be available since July 10<sup>th</sup>, 2007 at 11:00 GMT and remain available until undefined point of time.
- Furthermore, some of the contents associated with service “Music Channel” are available over the Interaction Channel so that the user can select those for downloading. In the example, “Music Video 2” and “Music Video 3” are such contents. Neither of these have defined availability time meaning these contents have been available since undefined point of time and will be available until undefined point of time for delivery as files over Interaction Channel.
- Note that Music News is available over both the Broadcast Channel as well as the Interaction Channel, while “Music Video 1”, “Music Video 2” and “Music Video 3” are only available over the interaction channel.
- Contents that are delivered over Broadcast Channel are defined as a purchasable item.
- Contents that are available over Interaction Channel are defined as another purchasable item.
- Note that the same content “Music News” is available on both Broadcast Channel and Interaction Channel at July 10<sup>th</sup>, 2007 at 11:00 GMT. While at July 10<sup>th</sup>, 2007 at 12:00 the Broadcast Channel starts to deliver next content, the “Music News” remains available on the Interaction Channel.

- Figure 38 above depicts the scenario on a timeline.

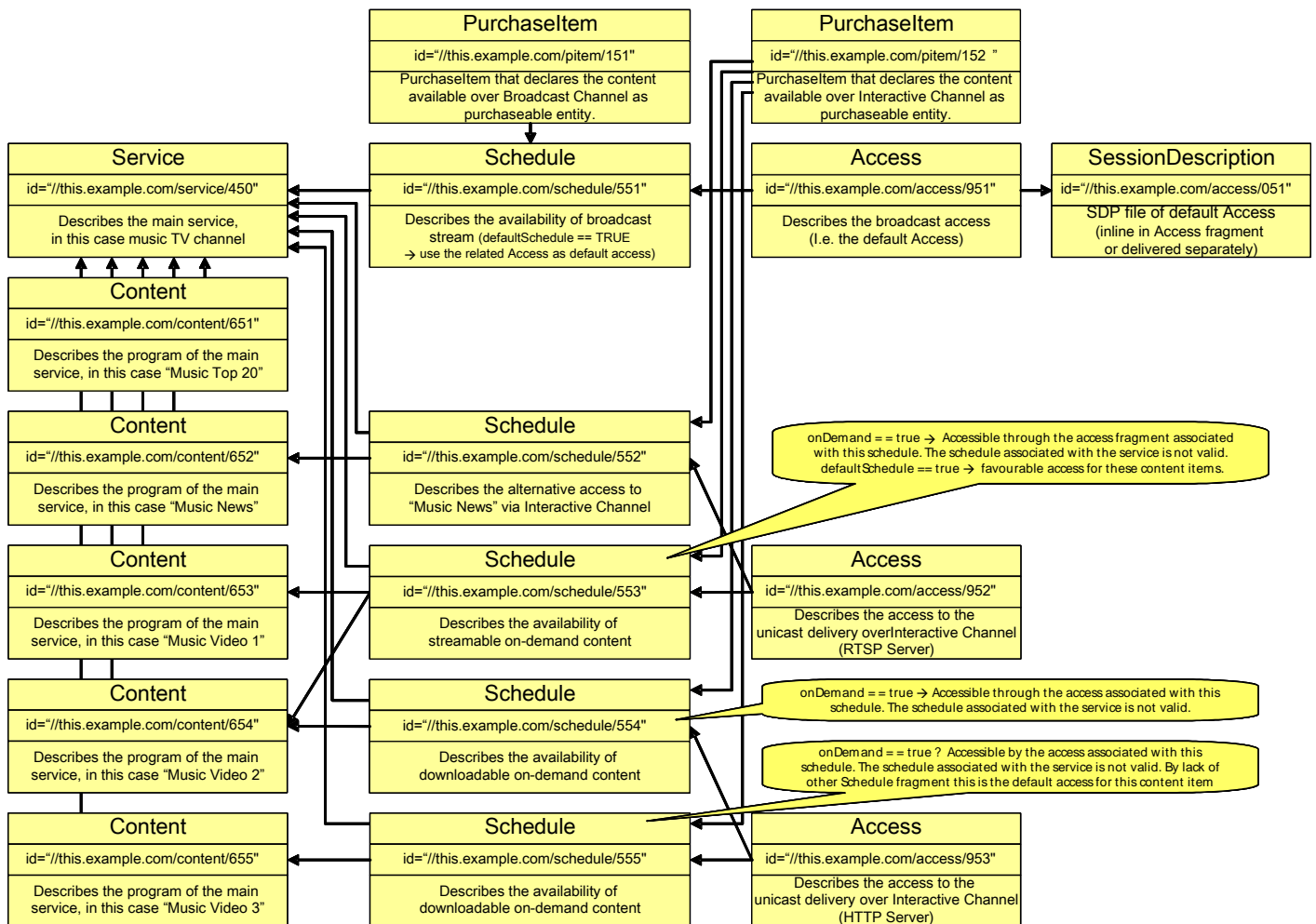


Figure 39: Example of Service Guide instantiation

The Service Guide is instantiated to support the scenario as follows:

- A single ‘Service’ fragment describes the “Music Channel”.
- For each of the contents “Music Top 20”, “Music News”, “Music Video 1”, “Music Video 2” and “Music Video 3” there are an individual ‘Content’ fragments describing the contents. The ‘StartTime’ and ‘EndTime’ within the ‘Content’ fragments declare the time windows when the corresponding contents are available (either over Broadcast Channel or over Interaction Channel).
- ‘Access’ fragment with identifier “//this.example.com/access/951” describes the access to “Music Channel” over Broadcast Channel. Within the ‘Access’ fragment the E2-element ‘Access.BroadcastServiceDelivery’ is instantiated to declare this access is over the Broadcast Channel. The attribute ‘Access.BroadcastServiceDelivery BDSType.type’ is set to “0 – IPDC over DVB-H” to denote that the delivery takes place over DVB-H bearer.
- ‘SessionDescription’ is instantiated as SDP description and associated with ‘Access’ fragment “//this.example.com/access/951” and hence describing the way to access the service “Music Channel” over the Broadcast Channel. Alternatively, the ‘SessionDescription’ could be instantiated as an inline element within the ‘Access’ fragment.
- Five ‘Schedule’ fragments are used to describe the availability of contents as follows:

- The ‘Schedule’ fragment with identifier “//this.example.com/schedule/551” schedules the content associated with “Music Channel” on the Broadcast Channel. This ‘Schedule’ fragment declares itself being the default ‘Schedule’ for “Music Channel”. This is done by setting the attribute ‘defaultSchedule’ in the ‘Schedule’ fragment to ‘true’. Consequently this declaration means that the associated ‘Access’ fragment “//this.example.com/access/951” will be the default access for the “Music Channel”. Note that this ‘Schedule’ fragment does not refer to any individual ‘Content’ fragments.
- The ‘Schedule’ fragment with identifier “//this.example.com/schedule/552” declares that:
  - the content “Music News” is available for the streaming delivery over the Interaction Channel.
  - this ‘Schedule’ fragment describes the alternative access to the “Music News”. This is done by associating ‘Access’ fragment “//this.example.com/access/952” with a ‘Schedule’ fragment without a ‘defaultSchedule’ attribute declared
  - the reference to “Music News” is instantiated as attribute ‘idRef’ under ‘ContentReference’ element to ‘Content’ fragment “//this.example.com/content/652”.
  - the ‘contentLocation’ attribute under the ‘ContentReference’ element contains the ‘abs\_path’ part of the RTSP ‘Request-URI’ used when “Music News” is to be streamed through the associated ‘Access’.
- The ‘Schedule’ fragment with identifier “//this.example.com/schedule/553” declares that:
  - the contents “Music Video 1” and “Music Video 2” are available for the streaming delivery over the Interaction Channel.
  - the ‘Access’ fragment “//this.example.com/access/952”, which describes the streaming delivery, is associated with ‘Schedule’ fragment through the reference within the ‘Access’ fragment.
  - This is the favourable schedule of the content item with identifier “//this.example.com/content/653” and “//this.example.com/content/654”. This is done by setting the attribute ‘defaultSchedule’ in the ‘Schedule’ fragment to ‘true’.
  - the references to two associated pieces of content are instantiated as individual ‘ContentReference’ elements with attribute ‘idRef’ to ‘Content’ fragments “//this.example.com/content/653” and “//this.example.com/content/654”.
  - the ‘contentLocation’ attribute under each ‘ContentReference’ contains the ‘abs\_path’ part of the RTSP ‘Request-URI’ used when “Music Video 1” and “Music Video 2” are to be streamed through the associated ‘Access’
  - it is part of the “on demand” service and the associated content is not accessed automatically through this ‘Schedule’ fragment by the terminal when this ‘Schedule’ fragment is valid. This is done by setting the ‘onDemand’ attribute to ‘true’.
  - it is not accessible through the main Schedule fragment with identifier “//this.example.com/schedule/551”. This is also done by setting the ‘onDemand’ attribute to ‘true’.
- The ‘Schedule’ fragment with identifier “//this.example.com/schedule/554” declares that:
  - the contents “Music Video 2” is available for the download delivery over the Interaction Channel.
  - the ‘Access’ fragment “//this.example.com/access/953”, which describes the download delivery, is associated with ‘Schedule’ fragment through the reference within the ‘Access’ fragment.
  - the reference to associated content is instantiated as attribute ‘idRef’ under ‘ContentReference’ elements to ‘Content’ fragments “//this.example.com/content/654”.
  - the ‘contentLocation’ attribute under ‘ContentReference’ contains the ‘abs\_path’ part of the HTTP URL used when “Music Video 2” is to be downloaded through the associated ‘Access’.
  - it is part of the “on demand” service and the associated content is not accessed automatically through this ‘Schedule’ fragment by the terminal when this ‘Schedule’ fragment is valid. This is done by setting the ‘onDemand’ attribute to ‘true’.

- it is not accessible through the main Schedule fragment with identifier “//this.example.com/schedule/551”. This is also done by setting the ‘onDemand’ attribute to ‘true’.
  - The ‘Schedule’ fragment with identifier “//this.example.com/schedule/555” declares that:
    - the contents “Music Video 3” is available for the download delivery over the Interaction Channel.
    - the ‘Access’ fragment “//this.example.com/access/953”, which describes the download delivery, is associated with ‘Schedule’ fragment through the reference within the ‘Access’ fragment.
    - the reference to associated content is instantiated as attribute ‘idRef’ under ‘ContentReference’ elements to ‘Content’ fragments “//this.example.com/content/655”.
    - the ‘contentLocation’ attribute under ‘ContentReference’ contains the ‘abs\_path’ part of the HTTP URL used when “Music Video 3” is to be downloaded through the associated ‘Access’.
    - it is part of the “on demand” service and the associated content is not accessed automatically through this ‘Schedule’ fragment by the terminal when this ‘Schedule’ fragment is valid. This is done by setting the ‘onDemand’ attribute to ‘true’.
    - it is not accessible through the main Schedule fragment with identifier “//this.example.com/schedule/551”. This is also done by setting the ‘onDemand’ attribute to true.
  - Each of the mentioned ‘Schedule’ fragments have a reference to ‘Service’ fragment “//this.example.com/service/450” associating these schedules with the service “Music Channel”.
- ‘Access’ fragment with identifier “bcast://this.operator/access/952” describes the access over the Interaction Channel to the contents associated with ‘Schedule’ fragments with identifiers “//this.example.com/schedule/552” and “//this.example.com/schedule/553”, i.e. the contents “Music News”, “Music Video 1” and “Music Video 2”. Within the ‘Access’ fragment, the E2-element ‘UnicastServiceDelivery’ is instantiated to declare this access is over the Interaction Channel and the attribute ‘type’ is set to “3 - RTSP” to indicate this is a streaming delivery for which the stream setup and negotiation is done using the RTSP protocol.
  - The ‘AccessServerURL’ of this fragment carries the scheme ‘rtsp://’ and the hostname/port of the server with which the terminal will establish the RTSP session.
  - From that server, the terminal will request the corresponding content through the RTSP method using a RTSP ‘Request-URI’ created from the values of the ‘AccessServerURL’ element and the attribute ‘contentLocation’ of element ‘ContentReference’ of ‘Schedule’ fragment
- ‘Access’ fragment with identifier “//this.example.com/access/953” describes the access over the Interaction Channel to the contents associated with ‘Schedule’ fragments with identifiers “//this.example.com/schedule/554” and “//this.example.com/schedule/555”, i.e. the contents “Music Video 2” and “Music Video 3”. Within the ‘Access’ fragment the E2-element ‘UnicastServiceDelivery’ is instantiated to declare this access is over the Interaction Channel and the attribute ‘type’ is set to “0 - HTTP” to indicate this is a download delivery.
  - The ‘AccessServerURL’ of this fragment carries the scheme ‘http://’ and the hostname/port of the server with which the terminal will establish the HTTP session.
  - From that server, the terminal will request the corresponding content through the HTTP GET method as addressed by the HTTP URL created from the values of the ‘AccessServerURL’ element and the attribute ‘contentLocation’ under ‘ContentReference’ of ‘Schedule’ fragment “//this.example.com/schedule/554”.
- ‘PurchaseItem’ fragment with identifier “//this.example.com/pitem/151” is refers to ‘Schedule’ fragment with identifier “//this.example.com/schedule/551”. This way the contents of “Music Channel” delivered over Broadcast Channel are defined as a purchasable item.
- ‘PurchaseItem’ fragment with identifier “//this.example.com/pitem/152” is refers to ‘Schedule’ fragments with identifiers “//this.example.com/schedule/552”, “bcast://this.operator/schedule/553”, “//this.example.com/schedule/554” and “//this.example.com/schedule/555”. This way the contents of “Music Channel” delivered over Interaction Channel are defined as another purchasable item.
- Figure 39 above depicts the necessary Service Guide fragments and their relations.

### I.3.2 Scenario of Service Guide with Broadcasted Stream Service alternatively available over the Interaction Channel.

As previously mentioned, a common use of the interaction channel in a hybrid broadcast/unicast Mobile TV system is to have the broadcasted content alternatively available over the interaction channel. For example, the interaction channel is used as a fallback delivery method for a stream service in case the terminal goes out-of-coverage of the broadcast channel. The application scenario in this section explains how the Service Guide can be instantiated to support such use, and to do so only slightly modifies the instantiation described in section I.3.1.

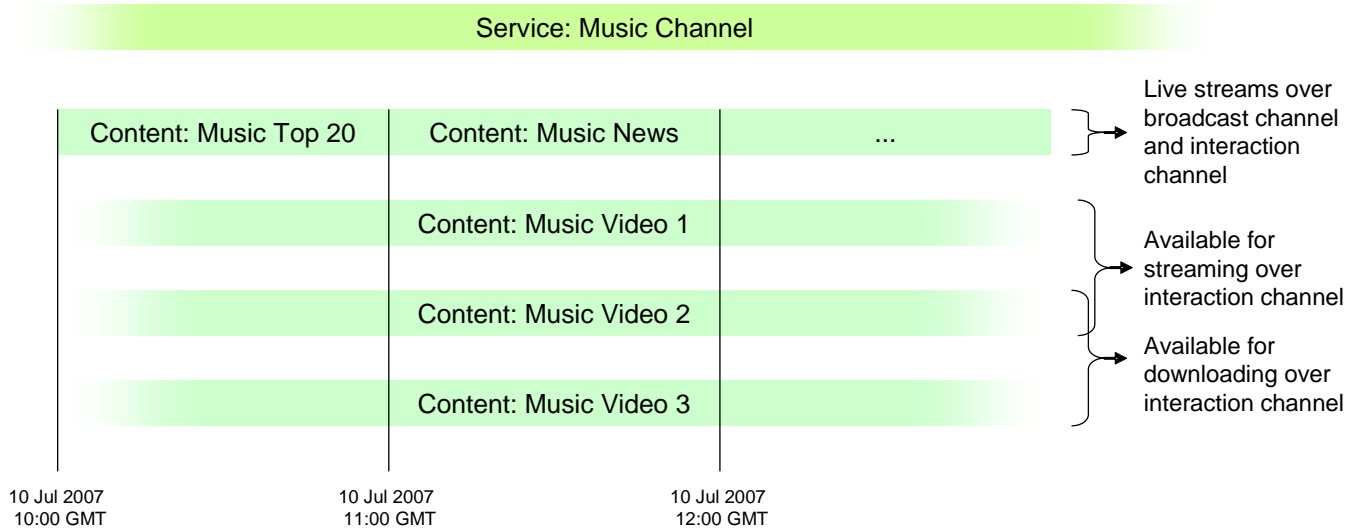
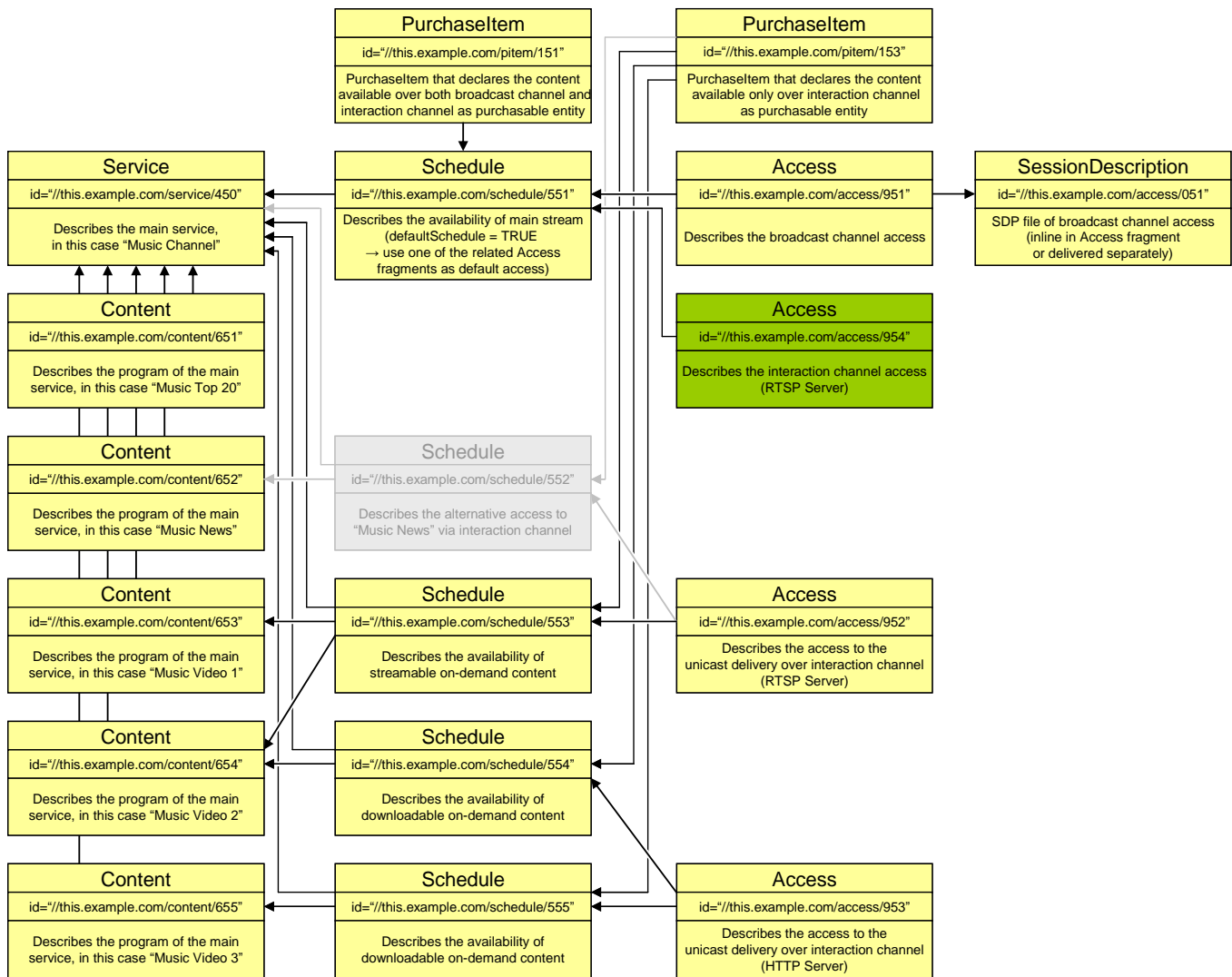


Figure 40: Service example on a timeline

The assumptions made in this scenario are the same as those made in section I.3.1 with the following exceptions:

- The live stream in which the “Music Channel” service delivers sequentially scheduled content consisting of individual programs such as “Music Top 20” and “Music News” is available both over the broadcast channel and interaction channel.
- The scenario described in section I.3.1 assumes that content “Music News” is available until an undefined point in time over the interaction channel. That is not the case in this scenario as illustrated in Figure 40 which only includes “Music News” in the sequentially scheduled live stream.
- Contents that are delivered in the live stream over broadcast channel and interaction channel are defined as a purchasable item.
- Contents that are available only over the interaction channel are defined as another purchasable item.
- Figure 40 above depicts the scenario on a timeline.



**Figure 41: Example of Service Guide instantiation (fragment in gray is removed compared to the Service Guide instantiation illustrated in Figure 39 , and fragments in green are added compared to the Service Guide instantiation illustrated in Figure 39)**

The instantiation illustrated in this scenario is the same as the one in section I.3.1 except for the following differences:

- ‘Access’ fragment with identifier “//this.example.com/access/954” has been added and describes alternative access over interaction channel for the sequentially scheduled live content in service “Music Channel” which is also available over the broadcast channel. Within the ‘Access’ fragment, the E2-element ‘UnicastServiceDelivery’ is instantiated to declare this access is over the interaction channel and the attribute ‘type’ is set to “3 - RTSP” to indicate this is a streaming delivery for which the stream setup and negotiation is done using the RTSP protocol.
  - The ‘AccessServerURL’ of this fragment carries the scheme ‘rtsp://’ and the hostname/port of the server with which the terminal will establish the RTSP session. Alternatively, a SDP identifying the streaming server and the content is designated by the ‘SDPRef’ element or embedded in the ‘SDP’ element of the ‘SessionDescription’ element of the ‘UnicastServiceDelivery’ element in the ‘Access’ fragment.
- ‘Schedule’ fragment with identifier “//this.example.com/schedule/551” schedules the content associated with “Music Channel” available over both the broadcast channel and interaction channel.

- This ‘Schedule’ fragment declares itself being the default ‘Schedule’ for “Music Channel”. This is done by setting the attribute ‘defaultSchedule’ in the ‘Schedule’ fragment to ‘true’. This attribute is set to ‘true’ since the instantiation includes multiple ‘Schedule’ fragments and at least two of them are referencing ‘Content’ or ‘Service’ fragments that overlap, i.e., that are valid at the same time (see introduction to section 5.1.2.2). Consequently this declaration means that the associated ‘Access’ fragments, “//this.example.com/access/951” and “//this.example.com/access/954”, will be the default access for the “Music Channel” service. Which of the ‘Access’ fragments to use is up to the terminal to decide (see section 5.8.1.1).
- Even if not applicable for this particular example of Service Guide instantiation, it should be noted that the ‘contentLocation’ attribute of the ‘ContentReference’ element in the ‘Schedule’ fragment can be instantiated even if multiple ‘Access’ fragments describing both interaction channel access and broadcast channel access are referencing the ‘Schedule’ fragment in question (see section 5.8.4.2). The reason why this note is not applicable for the instantiation in Figure 41 is that the ‘Schedule’ fragment with id “//this.example.com/schedule/551” is referencing the ‘Service’ fragment, and not a ‘Content’ fragment.
  - ‘Schedule’ fragment with identifier “//this.example.com/schedule/552” has been removed which reflects the assumption presented above regarding the availability of content “Music News”, i.e., content “Music News” is only available as part of the sequentially scheduled live content.

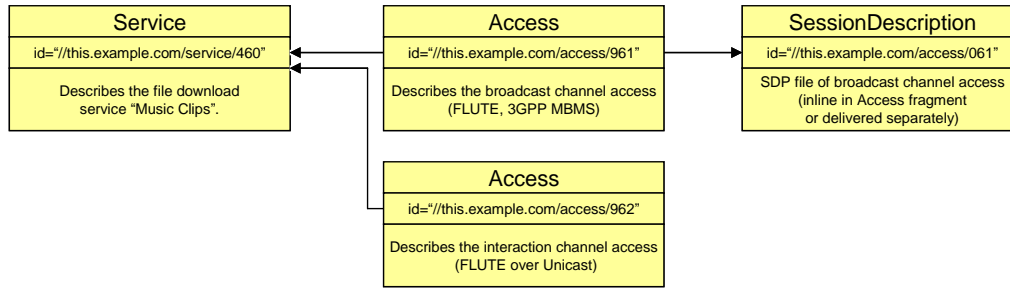
As seen in Figure 41, ‘PurchaseItem’ fragment with identifier “//this.example.com/pitem/151” refers to ‘Schedule’ fragment with identifier “//this.example.com/schedule/551” which means that the sequentially scheduled live contents in service “Music Channel” delivered over both the broadcast channel and interaction channel are defined as a purchasable item. Which access to use is up to the terminal to decide but the possible choices are reflected in the Service Guide instantiation by having ‘Access’ fragment with identifier “//this.example.com/access/951” and ‘Access’ fragment with identifier “//this.example.com/access/954” both referring to ‘Schedule’ fragment with identifier “//this.example.com/schedule/551”.

### I.3.3 Scenario of Service Guide with Broadcasted File Download Service alternatively available over the Interaction Channel

In a hybrid broadcast/unicast system it is possible for a file download service available over the broadcast channel to be alternatively available over the interaction channel. The interaction channel access may then, e.g., be used as a fallback delivery method in case the terminal goes out-of-coverage of the broadcast channel. For the broadcast channel access, the file download service is delivered using either FLUTE or ALC (see chapter 5 of [BCAST12-Distribution]), and for the interaction channel access, the file download service is delivered using either FLUTE (see section 5.5.1 of [BCAST12-Distribution]) or HTTP (see section 5.5.2 of [BCAST12-Distribution]). The application scenario in this section explains one example of how the Service Guide can be instantiated to enable broadcast/unicast hybrid scenarios.

In this example scenario the following is assumed

- There is a file download service called “Music Clips” that represents a service delivering music related contents. The users are able to access the “Music Clips” in two ways – either receiving the service over broadcast channel or accessing the contents over interaction channel. These two ways are uniformly represented as a single service on the user interface.
- For the broadcast channel, the “Music Clips” service is delivered over MBMS using FLUTE as the transport method.
- For the interaction channel, the “Music Clips” service is delivered using FLUTE as the transport method.
- The scheduling of the content delivered by the file download service is not visible in the Service Guide. This information is only available in the FDT instances delivered within the FLUTE session.



**Figure 42: Example of Service Guide instantiation for file download service available over both broadcast channel and interaction channel**

The Service Guide is instantiated to support the scenario as follows:

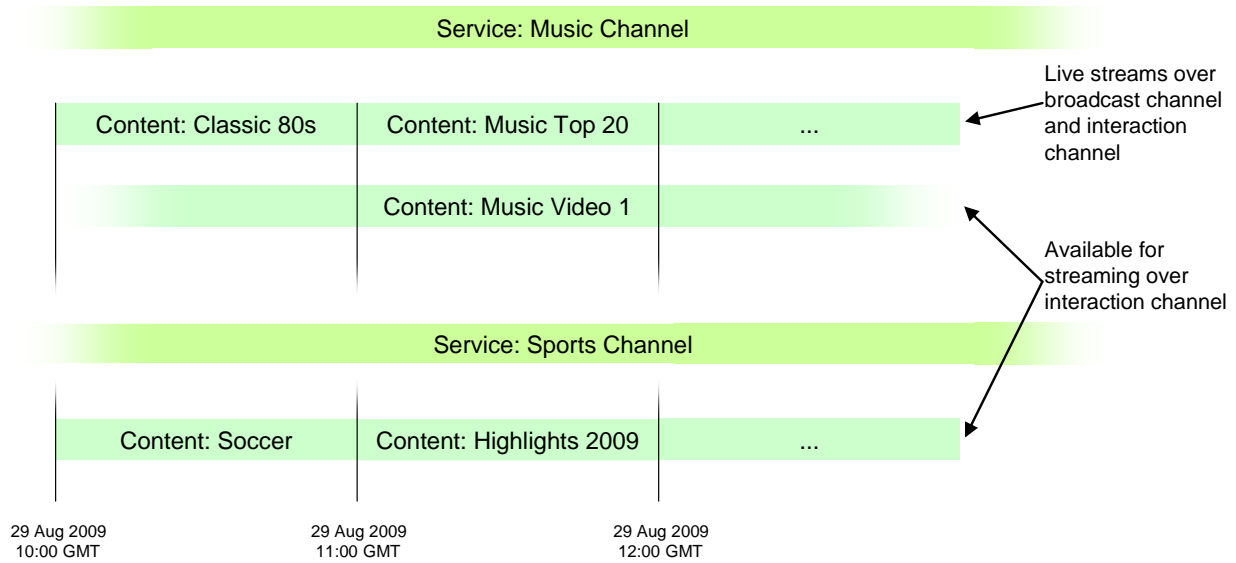
- A single ‘Service’ fragment describes the “Music Clips” service. The ‘ServiceType’ attribute is set to 5 (“File download services”).
- ‘Access’ fragment with identifier “//this.example.com/access/961” describes the access to “Music Clips” over broadcast channel. Element ‘Access.BroadcastServiceDelivery’ is instantiated to declare that this access is over the broadcast channel. The attribute ‘Access.BroadcastServiceDelivery.BDSType.type’ is set to “1 – 3GPP MBMS” to denote that the delivery takes place over MBMS bearer and the ‘Access.BroadcastServiceDelivery.FileDescription’ element is omitted to signal that FLUTE is used as the transport method.
- ‘SessionDescription’ fragment with identifier “//this.example.com/access/061” is instantiated as SDP description and associated with ‘Access’ fragment “//this.example.com/access/961” and hence describing the way to access the service “Music Clips” over the broadcast channel. Alternatively, the ‘SessionDescription’ could be instantiated as an inline element within the ‘Access’ fragment.
- ‘Access’ fragment with identifier “//this.example.com/access/962” describes alternative access to “Music Clips” over interaction channel. Element ‘Access.UnicastServiceDelivery’ is instantiated to declare this access is over the interaction channel. The attribute ‘Access.UnicastServiceDelivery.type’ is set to “6 - FLUTE over Unicast” to denote that the interaction channel delivery takes place using FLUTE.
  - The ‘AccessServerURL’ of this fragment carries the scheme ‘rtsp://’ and the hostname/port of the server with which the terminal will establish the RTSP session.
- Figure 42 above depicts the necessary Service Guide fragments and their relations.

## I.4 Example for Hybrid Broadcast/Unicast Service Guide Delivery

In a hybrid broadcast/unicast system it is possible for a Service Guide to be delivered solely over broadcast channel, solely over interaction channel, or over both broadcast channel and interaction channel. See section I.2 for more information about the possibilities to deliver a Service Guide in a hybrid broadcast/unicast environment.

This section contains an example of hybrid broadcast/unicast Service Guide delivery where the delivery takes place over both broadcast channel and interaction channel, and where the initial SGDD is received over the broadcast channel. Furthermore, the example also assumes that the Service Guide delivered over the interaction channel is a superset of the broadcasted Service Guide.





**Figure 43: Services example on a timeline**

In this example scenario the following is assumed

- There is a service called “Music Channel” that represents a mobile TV / video service delivering music related contents. The users are able to access the main content of “Music Channel” in two ways – either over broadcast channel or over interaction channel. Some additional content like “Music Video 1” is only accessible over the interaction channel.
- There is a service called “Sports Channel” that represents a mobile TV / video service delivering sports related contents. This service is only accessible over the interaction channel.
- Figure 43 above depicts the scenario on a timeline.

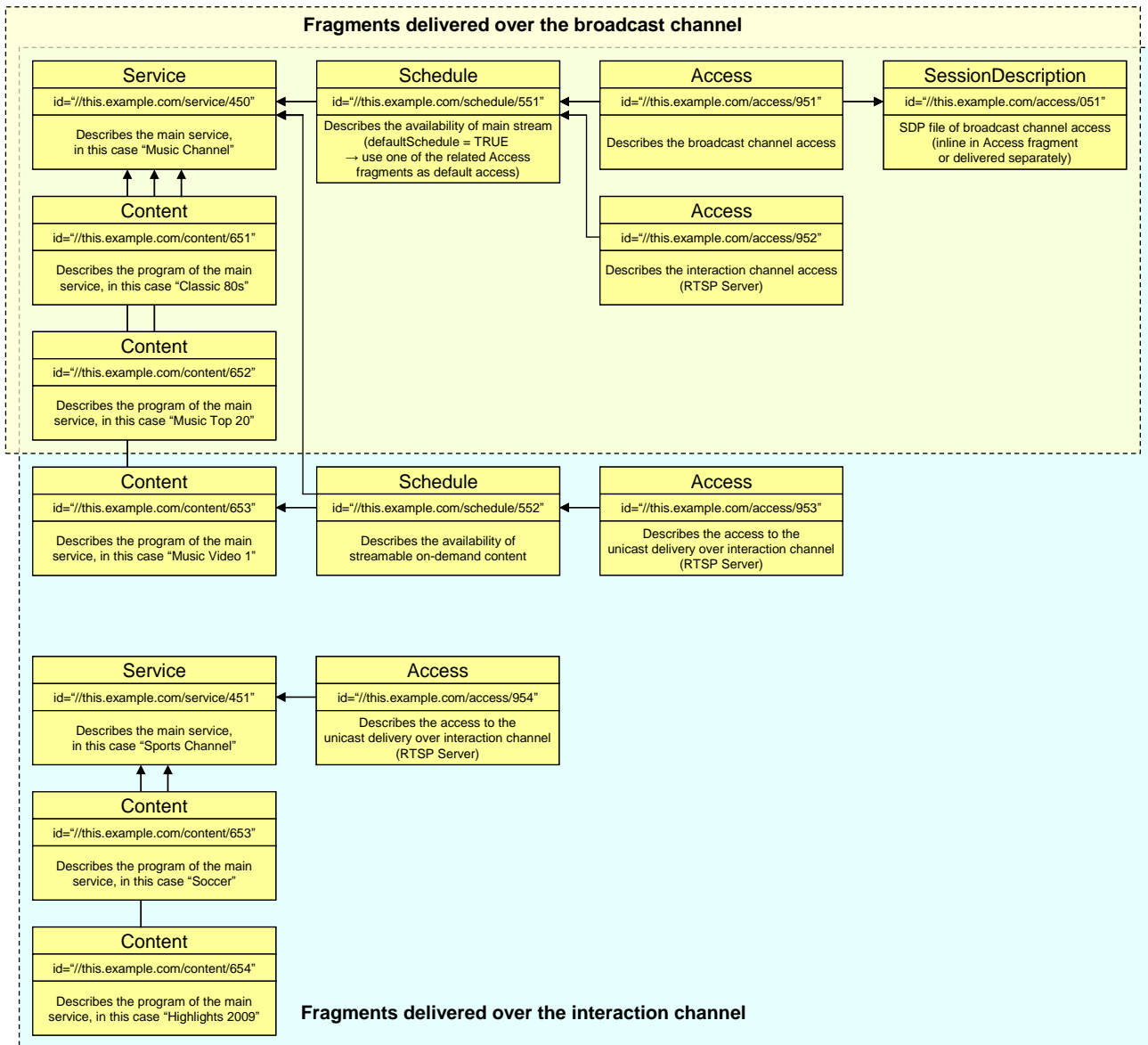


Figure 44: Example of Service Guide instantiation

The Service Guide is instantiated to support the scenario as follows:

- ‘Service’ fragment with identifier ‘//this.example.com/service/450’ describes the service “Music Channel”.
  - For each of the contents “Classic 80s”, “Music Top 20”, and “Music Video 1” there are an individual ‘Content’ fragment describing the contents.
  - ‘Schedule’ fragment with identifier “//this.example.com/schedule/551” schedules the content associated with “Music Channel” available over both the broadcast channel and interaction channel, i.e., schedules content “Classic 80s” and “Music Top 20”. See section I.3.2 for more information how to instantiate fragments in a hybrid broadcast/unicast scenario.
  - ‘Content’ fragment with identifier “//this.example.com/content/653”, ‘Schedule’ fragment with identifier “//this.example.com/schedule/552”, and ‘Access’ fragment with identifier “//this.example.com/access/953” are only available for delivery over the interaction channel and describes the additional on-demand content “Music Video 1”.

- ‘Service’ fragment with identifier ‘//this.example.com/service/451’ describes the service “Sports Channel” which is only available for delivery over the interaction channel.
  - For each of the contents “Soccer” and “Highlights 2009” there are an individual ‘Content’ fragment describing the contents. These contents correspond to ‘Content’ fragment with identifier ‘//this.example.com/content/653’ and ‘Content’ fragment with identifier ‘//this.example.com/content/654’, respectively.
  - ‘Access’ fragment with identifier “//this.example.com/access/954” describes the access to “Sports Channel” which is over the interaction channel.

Figure 44 above depicts the Service Guide fragments and their relations. It also illustrates that all fragments are available for delivery over the interaction channel while a subset of the fragments are available for delivery over the broadcast channel. All fragments which describe content only streamed over the interaction channel are also only delivered over the interaction channel. This setup might for example be used in a deployment where there are some terminals without interaction channel access and some terminals that support the interaction channel. Terminals without interaction channel access will then not receive any fragments which are only applicable for interaction channel content. Although, ‘Access’ fragment with identifier ‘//this.example.com/access/952’ is in this example available over both broadcast channel and interaction channel since it might be beneficial for a hybrid terminal to have it readily available in case broadcast coverage is lost.

The example explained in this section assumes that a terminal receives SGDD over broadcast channel and detects the existence of fragments available for delivery over interaction channel from the instantiation of the E1 element ‘SGEntryPoints’.

Name	Type	Example	Data Type
ServiceGuideDeliveryDescriptor	E		
SGEntryPoints	E1		
id	A	1234	unsignedInt
BSMSelector	E2		
idRef	A	//bsm001 <sup>1</sup>	anyURI
SGEntryPoint	E2		
<i>BroadcastServerSession</i>	E3	-- NOT INSTANTIATED --	
<i>ipAddress</i>	A	-- NOT INSTANTIATED --	string
<i>port</i>	A	-- NOT INSTANTIATED --	unsignedShort
<i>srcIpAddress</i>	A	-- NOT INSTANTIATED --	string
<i>transmissionSessionID</i>	A	-- NOT INSTANTIATED --	unsignedShort
<i>BDSType</i>	E4	-- NOT INSTANTIATED --	complexType
<i>type</i>	A	-- NOT INSTANTIATED --	unsignedByte
<i>version</i>	A	-- NOT INSTANTIATED --	string
<i>BDSSpecificEntryPointInfo</i>	E5	-- NOT INSTANTIATED --	Abstract complexType
UnicastServerURL	E3		
relationOfICWithBC	A	3	unsignedByte
url	A	http://provider.com/bcast-service-guide	anyURI
UnicastType	E4	0	unsignedByte
version	A	Rel-8	string
DescriptorEntry	E1		
Transport	E2		
ipAddress	A	232.0.0.2	string

<sup>1</sup> Value “//bsm001” identifies a specific BSM in the ‘BSMList’ element located in the SGDD. However, the instantiation of the ‘BSMList’ element is not illustrated in this example in order to ease the visibility of the other elements.

port	A	8765	unsignedShort
srcIpAddress	A	1.2.3.4	string
transmissionSessionID	A	456	unsignedShort
hasFDT	A	TRUE	boolean
<i>AlternativeAccessURL</i>	<i>E2</i>	-- NOT INSTANTIATED --	<i>anyURI</i>
<i>ServiceGuideDeliveryUnit</i>	<i>E2</i>		
transportObjectID	A	543	positiveInteger
contentLocation	A	urn:oma:bcast:sgdu:101	anyURI
validFrom	A	24 Aug 2009, 08:00 GMT in NTP	unsignedInt
validTo	A	31 Aug 2009, 08:00 GMT in NTP	unsignedInt
Fragment	E3		
transportID	A	1001	unsignedInt
id	A	//this.example.com/service/450	anyURI
version	A	3	unsignedInt
fragmentEncoding	A	0	unsignedByte
fragmentType	A	1	unsignedByte
<i>isAvailableOverIC</i>	<i>A</i>	-- NOT INSTANTIATED --	<i>Boolean</i>
...	...	...	...

**Table 12: Example of SGDD. Elements and attributes written in *italics* are not instantiated for this example but are include in the table since their absence is also commented in the bullets below.**

The SGDD is instantiated to support the scenario as follows:

- E1 element ‘SGEntryPoints’ is instantiated to signal additional Service Guide entry points
  - E2 element ‘BSMSelector’ is instantiated and associates the Service Guide entry point with BSM identified by “//bsm001”. Note that the corresponding instantiation of ‘BSMList’ element is not illustrated in Table 12 in order to ease the visibility of the other elements.
  - E2 element ‘SGEntryPoint’ is instantiated to signal the existence and relationship between Service Guide delivered over the broadcast channel and the Service Guide delivered over the interaction channel. Sub-element ‘BroadcastServerSession’ is used for signaling the existence of Service Guide entry point over broadcast channel and sub-element ‘UnicastServerURL’ is used for signaling the existence of Service Guide entry point over interaction channel. The relationship between these entry points is signaled in the ‘relationOfICWithBC’ attribute of the ‘UnicastServerURL’ sub-element as explained in the following sub-bullets.
    - E3 element ‘BroadcastServerSession’ is not instantiated to signal that it refers to the entry point (i.e., Announcement Channel) over which the current SGDD has been delivered. As previously mentioned, the example described in this section assumes that the initial SGDD is received over the broadcast channel.
    - E3 element ‘UnicastServerURL’ is instantiated to signal the existence of fragments available over the interaction channel.
      - Attribute ‘relationOfICWithBC’ is set to value 3 to signal that the fragments delivered over the broadcast channel are a subset of the fragments available for delivery over the interaction channel from the URL specified in the ‘url’ attribute.
      - Attribute ‘url’ specifies “http://provider.com/bcast-service-guide” as the entry point to Service Guide acquisition over interaction channel.
- E1 element ‘DescriptorEntry’ is used for signaling a set of fragments. Only the sub-elements of ‘DescriptorEntry’ which are important to the scenario described in this section are described in the following bullets.

- E2 element ‘AlternativeAccessURL’ is not allowed to be instantiated since ‘SGEntryPoints’ element and ‘UnicastServerURL’ element are instantiated as described above. See the description of ‘AlternativeAccessURL’ element in section 5.4.1.5.2 for more information.
- Attribute ‘isAvailableOverIC’ of the ‘Fragment’ element is not allowed to be instantiated since the Service Guide delivered over the interaction channel is a superset of the Service Guide delivered over the broadcast channel, as specified by the ‘relationOfICWithBC’ attribute above (value 3). This attribute is only used for situations where the Service Guide delivered over the interaction channel complements the Service Guide delivered over the broadcast channel and the ‘relationOfICWithBC’ attribute therefore instead has the value 1.

Taking into account the scenario assumptions listed above, one possible strategy for a terminal to receive the hybrid broadcast/unicast Service Guide is outlined below.

1. The terminal tunes to the broadcast channel and discovers the Service Guide Announcement Channel as explained in section 6.1.1
2. The SGDD described in Table 12 is received from the Service Guide Announcement Channel.
3. The terminal parses and interprets the SGDD from which the existence of a Service Guide entry point for interaction channel access is detected from E3 element ‘UnicastServerURL’. The ‘relationOfICWithBC’ attribute specifies that the Service Guide distributed over interaction channel is a superset of the Service Guide delivered over broadcast channel. If the terminal has interaction channel access, it might in this situation use one of the following strategies for Service Guide retrieval.
  - Receive all possible fragments from the broadcast channel and receive all additional fragments from interaction channel. For retrieving fragments available only over the interaction channel, the terminal can use the following conjunction of key-value pairs in a request sent to the URL specified by the ‘url’ sub-element of the ‘UnicastServerURL’ element:

*type=sgdu&SGExclusivelyOverIC=true*

For this example, the response to such a request will contain all fragments illustrated in Figure 44 which are only delivered over interaction channel. This includes ‘Content’ fragment with identifier ‘//this.example.com/content/653’, ‘Schedule’ with identifier ‘//this.example.com/schedule/552’, and ‘Access’ fragment with identifier ‘//this.example.com/access/953’ as well as ‘Service’ fragment with identifier ‘//this.example.com/service/451’ and its related ‘Content’ fragments and ‘Access’ fragment.

- Receive all possible fragments from the broadcast channel, receive all ‘Service’ fragments available over interaction channel, and then receive additional fragments from interaction channel when needed. For retrieving ‘Service’ fragments available only over the interaction channel, the terminal can use the following conjunction of key-value pairs in a request sent to the URL specified by the ‘url’ sub-element of the ‘UnicastServerURL’ element:

*type=sgdu&fragmentType=1&SGExclusivelyOverIC=true*

For this example, the response to such a request will contain ‘Service’ fragment with identifier ‘//this.example.com/service/451’. The terminal is now aware of all services that are available. The terminal can then retrieve additional fragments when needed. For example, if the user selects service “Music Channel”, he/she might also be interested in the “Music Video 1” on-demand content and any other content available over the interaction channel that is related to the “Music Video 1” service. To receive such fragments, the terminal can use the following conjunction of key-value pairs in a request sent to the URL specified by the ‘url’ sub-element of the ‘UnicastServerURL’ element:

*type=sgdu&globalServiceID=<globalServiceID>&all=true&SGExclusivelyOverIC=true*

Note: <globalServiceID> is specified in attribute ‘globalServiceID’ of the ‘Service’ fragment with identifier ‘//this.example.com/service/450’.

For this example, the response to such a request will contain ‘Content’ fragment with identifier ‘//this.example.com/content/653’, ‘Schedule’ with identifier ‘//this.example.com/schedule/552’, and ‘Access’ fragment with identifier ‘//this.example.com/access/953’.

- Note that for both cases above, the terminal can retrieve all fragments over the interaction channel in case broadcast coverage is lost. The terminal can then go back to receive fragments over the broadcast channel when coverage has been reacquired. The choice of access method might also be based on factors such as service provider preferences and user preferences.
4. Since the Service Guide delivered over the interaction channel is a superset of the Service Guide delivered over the broadcast channel, the terminal can use the terminal-based way or system-based way, described in section 5.5.2, for Service Guide update and management over the interaction channel. By monitoring the changes in Service Guide Delivery Descriptors delivered over the interaction channel, the terminal can monitor changes in the entire Service Guide.

As previously mentioned, this example assumes that the SGDD is initially received over the broadcast channel. Note that it is also possible for the initial SGDD to be received over the interaction channel and the broadcasted Service Guide entry point to be derived from that SGDD.