Mobile Broadcast Services

Approved Version 1.0 – 12 Feb 2009

Open Mobile Alliance

OMA-TS-BCAST_Services-V1_0-20090212-A

Contents

Figures

Tables

1. Scope

This specification, together with the other specification comprising the Mobile Broadcast Services Enabler (BCAST 1.0), define a technological framework and specify globally interoperable technologies for the generation, management and distribution of mobile broadcast services over different broadcast distribution systems. The complete list of the specifications for BCAST 1.0 is defined in the Enabler Release Definition of BCAST 1.0 [BCAST10-ERELD]. This enabler suite includes specifications for the following functions: Service Guide; Service and Content protection; File and Stream distribution; Terminal Provisioning; Service Provisioning; Notifications; and; Service Interaction. In addition, a specification is provided for Roaming, Mobility and Charging. Adaptations to specific broadcast distribution systems (3GPP/MBMS, 3GPP2/BCMCS and “IP Datacast over DVB-H”) are specified in the Adaptation Specification documents.

Overall, the scope of the BCAST 1.0 enabler is service layer technologies. Thus, all specifications address the protocol layers on top of the radio bearer level. Furthermore, a common nominator for all the BCAST 1.0 technologies is that they are based on Internet Protocol (IP) and technologies related to IP. This scoping applies to all features and functionalities specified in BCAST 1.0.

The following functions are included in this specification: Service Provisioning; Terminal Provisioning; Interaction, Personalization and Support for User-Based Profiles and Preferences; Security and Privacy; Charging; Mobility; Broadcast Roaming; Notification; and; Location Information. Further, this document provides mappings between the BCAST 1.0 interfaces as defined in BCAST Architecture [BCAST10-Architecture] and the various BCAST 1.0 Technical Specifications.

2. References

2.1 Normative References

[3GPP TS 22.022] “Personalisation of GSM Mobile Equipment (ME); Mobile functionality specification”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 22.022,
URL: http://www.3gpp.org/

[3GPP TS 23.003] “Numbering, addressing and identification”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 23.003,
URL: http://www.3gpp.org/

[3GPP TS 23.060] “General Packet Radio Service (GPRS); Service description; Stage 2”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 23.060,
URL: http://www.3gpp.org

[3GPP TS 24.008] “Mobile radio interface Layer 3 specification; Core network protocols; Stage 3”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 24.008, Release 6,
URL: http://www.3gpp.org/

[3GPP TS 25.413] “UTRAN Iu interface RANAP signaling”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 25.413, Release 6,
URL: http://www.3gpp.org/

[3GPP TS 26.245] “Packet switched Streaming Service (PSS);Timed text format”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 26.245, Release 6,
URL: http://www.3gpp.org/

[3GPP TS 26.246] “Transparent end-to-end Packet-switched Streaming Service (PSS); 3GPP SMIL language profile”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 26.246,
URL: http://www.3gpp.org/

[3GPP TS 26.346 v7] “Multimedia Broadcast/Multicast Service (MBMS), Protocols and codecs (Release 7)”, Technical Specification Group Services and System Aspects, 3rd Generation Partnership Project, 3GPP TS 26.346,
URL: http://www.3gpp.org/

[3GPP TS 33.246] “3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)”, 3rd Generation Partnership Project, Technical Specification 3GPP TS 33.246,
URL: http://www.3gpp.org/

[3GPP2 C.S0050] “3GPP2 File Formats for Multimedia Services”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0050,
URL: http://www.3gpp2.org/

[3GPP2 C.S0072] “Mobile Station Equipment Identifier (MEID) Support for CDMA 2000 Spread Spectrum Systems, Revision 0”, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0072,
URL: http://www.3gpp2.org/

[3GPP2 X.S0022-A] “Broadcast and Multicast Service in cdma2000 Wireless IP Network”, Release A, 3rd Generation Partnership Project 2, Technical Specification 3GPP2 X.S0022-A,
URL: http://www.3gpp2.org/

[BCAST10-BCMCS-Adaptation] "Broadcast Distribution System Adaptation – 3GPP2/BCMCS", Open Mobile Alliance™, OMA-TS-BCAST_BCMCS_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-Architecture] "Mobile Broadcast Services Architecture", Open Mobile Alliance™, OMA-AD- BCAST-V1_0,
http://www.openmobilealliance.org/

[BCAST10-DDF-BCAST-MO] "Mobile Broadcast Services – DDF of BCAST Management Object", Open Mobile Alliance™, OMA-SUP-MO_oma_bcast-V1_0,

URL: http://www.openmobilealliance.org/

[BCAST10-Distribution] "File and Stream Distribution for Mobile Broadcast Services ", Open Mobile Alliance™, OMA-TS-BCAST_Distribution-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-DVBH-IPDC-Adaptation] "Broadcast Distribution System Adaptation – IPDC over DVB-H", Open Mobile Alliance™, OMA-TS-BCAST_DVB_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-ERELD] "Enabler Release Definition for Mobile Broadcast Services", Open Mobile Alliance™, OMA-ERELD-BCAST-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-MBMS-Adaptation] "Broadcast Distribution System Adaptation – 3GPP/MBMS", Open Mobile Alliance™, OMA-TS-BCAST_MBMS_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-Requirements] "Mobile Broadcast Services Requirements", Open Mobile Alliance™, OMA-RD-BCAST-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-ServContProt] "Service and Content Protection for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_SvcCntProtection-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-Services] "Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_Services-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-SG] "Service Guide for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_ServiceGuide-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-InteractivityMedia] "Mobile Broadcast Services – XML Schema for InteractivityMediaDocument", Open Mobile Alliance™, OMA-SUP-XSD_bcast_si_interactivitymedia-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-orderqueries] "Mobile Broadcast Services – XML Schema for Service Provisioning Order Queries", Open Mobile Alliance™, OMA-SUP-XSD_bcast_pr_orderqueries-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-Roaming-backend] "Mobile Broadcast Services – XML Schema for Roaming Messages – Backend ", Open Mobile Alliance™, OMA-SUP-XSD_bcast_roaming_backend-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-Roaming-frontend] "Mobile Broadcast Services – XML Schema for Roaming Messages – Frontend", Open Mobile Alliance™, OMA-SUP-XSD_bcast_roaming_frontend-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-Userpreference] "Mobile Broadcast Services – XML Schema for User Preferences ", Open Mobile Alliance™, OMA-SUP-XSD_bcast_pr_userpreference-V1_0,
URL: http://www.openmobilealliance.org/

[DMBOOT] "OMA Device Management Bootstrap, Version 1.2". Open Mobile Alliance™, .
OMA-TS-DM_Bootstrap-V1_2.
URL: http://www.openmobilealliance.org/

[DMDDFDTD] "OMA DM Device Description Framework DTD, Version 1.2". Open Mobile Alliance™, .
OMA-SUP-dtd_dm_ddf-v1_2.
URL: http://www.openmobilealliance.org/

[DMNOTI] "OMA Device Management Notification Initiated Session, Version 1.2". Open Mobile Alliance™. OMA-DM_Notification-V1_2. .
URL: http://www.openmobilealliance.org/

[DMPRO] "OMA Device Management Protocol, Version 1.2". Open Mobile Alliance™, .
OMA-TS-DM_Protocol-V1_2.
URL: http://www.openmobilealliance.org/

[DMREPU] "OMA Device Management Representation Protocol, Version 1.2", .
Open Mobile Alliance™. OMA-TS-DM_RepPro-V1_2.
URL: http://www.openmobilealliance.org/

[DMSEC] "OMA Device Management Security, Version 1.2". Open Mobile Alliance™, .
OMA-TS-DM_Security-V1_2.
URL: http://www.openmobilealliance.org/

[DMSTDOBJ] “OMA Device Management Standardized Objects, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_StdObj-V1_2.
URL: http://www.openmobilealliance.org/

[DMTND] “OMA Device Management Tree and Description, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_TND-V1_2.
URL: http://www.openmobilealliance.org/

[DMTNDS] “OMA Device Management Tree and Description Serialization, Version 1.2”. Open Mobile Alliance™. OMA-TS-DM_TNDS-V1_2.
URL: http://www.openmobilealliance.org/

[DRM20-Broadcast-Extensions] "OMA DRM v2.0 Extensions for Broadcast Support", Open Mobile Alliance™, OMA-TS-DRM-XBS-V1_0,
URL: http://www.openmobilealliance.org/

[DRMDRM-v2.0] “DRM Specification V2.0”, Open Mobile Alliance™, OMA-DRM-DRM-V2_0,
URL: http://www.openmobilealliance.org/

[ERELDSC] “Enabler Release Definition for SyncML Common Specifications, version 1.2”. Open Mobile Alliance™. OMA-ERELD-SyncML-Common-V1_2.
URL: http://www.openmobilealliance.org/

[HTML4.01] “HTML 4.01 Specification”, W3C Recommendation 24 December 1999,
URL: http://www.w3.org/TR/html401/

[IOPPROC] “OMA Interoperability Policy and Process”, Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1,
URL: http://www.openmobilealliance.org/

[ITU-MCC] “List of Mobile Country or Geographical Area Codes”, ITU-T Telecommunication Standardization Sector of ITU Complement To ITU-T Recommendation E.212 (05/2004),

URL: http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212A-2007-PDF-E.pdf

Note: This List will be updated regularly by numbered series of amendments published in ITU Operational Bulletin. For the latest version see:

URL: http://www.itu.int/itu-t/bulletin/annex.html

[MMSCONF] “MMS Conformance Document 1.3”, Open Mobile AllianceOpen Mobile Alliance™, □. OMA-MMS-CONF-1_3.doc.
URL: http://www.openmobilealliance.org/

[MMSTEMP] “MMS Message Template Specification 1.3”, Open Mobile Alliance™, Open Mobile Alliance□. OMA-MMS-TEMP-1_3.doc.
URL: http://www.openmobilealliance.org/

[OMA Charging AD] “Charging Architecture”, Open Mobile AllianceOpen Mobile Alliance™, OMA-AD-Charging-V1_0-20060511-D,
URL: http://www.openmobilealliance.org/

[OMA DM] “Enabler Release Definition for OMA Device Management v1.2”, OMA-ERELD-DM-V1_2_0,
URL: http://www.openmobilealliance.org/

[OMA FUMO] "OMA Enabler Release Definition for Firmware Update Management Object v1.0", Open Mobile Alliance™, OMA-ERELD-FUMO-V1_0,
URL: http://www.openmobilealliance.org/

[OMA MLP] “Mobile Location Protocol”, Open Mobile AllianceOpen Mobile Alliance™TM, OMA-TS-MLP-V3_2
URL: http://www.openmobilealliance.org/

[RFC 1951] “DEFLATE Compressed Data Format Specification version 1.3”, P. Deutsch, May 1996,
URL:http://www.ietf.org/rfc/rfc1951.txt

[RFC 1952] “ZIP file format specification version 4.3”, P. Deutsch, May 1996,
URL:http://www.ietf.org/rfc/rfc1952.txt

[RFC 2048] “Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures”, N. Freed, J. Klensin, J. Postel, November 1996,
URL: http://www.ietf.org/rfc/rfc2048.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

[RFC 2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997,
URL: 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

[RFC 2616] IETF RFC 2616, “Hypertext Transfer Protocol -- HTTP/1.1”,
URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC 2822] RFC 2822, “Internet Message Format”, P. Resnick, Ed. April 2001,
URL: http://www.ietf.org/rfc/rfc2822.txt.

[RFC 2865] “Remote Authentication Dial In User Service (RADIUS)”, The Internet Engineering Task Force RFC 2865,
URL: http:// www.ietf.org/

[RFC 3261] “SIP: Session Initiation Protocol”, Rosenberg, J. et al, June 2002,
URL: http://www.ietf.org/rfc/rfc3261.txt

[RFC 3966] “The tel URI for Telephone Numbers”, Schulzrinne, H., December 2004,
URL: http://www.ietf.org/rfc/rfc3966.txt

[RFC4234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. October 2005,
URL:http://www.ietf.org/rfc/rfc4234.txt

[SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures,
URL:http://www.openmobilealliance.org/

[SSL30] “SSL 3.0 Specification”, Netscape Communications, November 1996,
URL: http://wp.netscape.com/eng/ssl3/draft302.txt

[URI-Schemes] “URI Schemes for the Mobile Applications Environment”, Version 1.0, Open Mobile Alliance™,
URL: http://www.openmobilealliance.org/

[XHTMLMP11] "XHTML Mobile Profile 1.1", Open Mobile AllianceOpen Mobile Alliance™. OMA-WAP-XHTMLMP-V1_1.
URL: http://www.openmobilealliance.org/

[XML] Extensible Markup Language (XML) 1.1, W3C Recommendation 04 February 2004, edited in place 15 April 2004.
URL: http://www.w3.org/TR/xml11

[XMLSchema] XML Schema,
URL: http://www.w3.org/XML/Schema

2.2 Informative References

[BCAST10-Architecture] "Mobile Broadcast Services Architecture", Open Mobile Alliance™, OMA-AD- BCAST-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-ERELD] "Enabler Release Definition for Mobile Broadcast Services”, Open Mobile Alliance™, OMA-ERELD-BCAST-V1_0,
URL: http://www.openmobilealliance.org/

[DMACMO] “White Paper on Provisioning Objects”. Open Mobile Alliance™. OMA-WP-AC_MO.
URL: http://www.openmobilealliance.org/

[ETSI 102 470] ETSI TS 102 470 v1.1.1 (2006-06), “Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Program Specific Information (PSI)/Service Information (SI)”,
URL: http://portal.etsi.org

[OMADICT] “Dictionary for OMA Specifications”, Version x.y, Open Mobile Alliance™,
OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org/

[RFC 4281] “The Codecs Parameter for "Bucket" Media Types”, R. Gellens, D. Singer, P. Frojdh, November 2005,

URL:http://www.ietf.org/rfc/rfc4281.txt

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 itssub-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

Broadcast Roaming Broadcast Roaming is the ability of a user to receive broadcast services from a Mobile Broadcast Service Provider different from the Home Mobile Broadcast Service Provider with which the user has a contractual relationship.

Broadcast Service A Broadcast Service is a “content package” suitable for simultaneous distribution to many recipients (potentially) without knowing the recipient. Either each receiver has similar receiving devices or the content package includes information, which allows the client to process the content according to his current conditions.

Examples of Broadcast Services are:

- pure Broadcast Services:
  - mobile TV
  - mobile newspaper
  - mobile file downloading (clips, games, SW upgrades, other applications, applications)
- combined broadcast/interactive Broadcast Services:
  - mobile TV for file downloading with voting
  - betting Broadcast Services
  - auction Broadcast Services
  - trading Broadcast Services

Broadcast Service Area The geographical or logical area in which a Broadcast Service is distributed.

CSIM
Acronym for 'cdma2000 Subscriber Identify Module', corresponding to an application defined in [3GPP2 C.S0065] residing on the UICC to register services provided by 3GPP2 mobile networks with the appropriate security.

Home Mobile Broadcast Service Provider
The Mobile Broadcast Service Provider with which the user has a subscription. Typically a user has one Home Mobile Broadcast Service Provider. However, the user may also have no Home Mobile Broadcast Service Provider or several Home Mobile Broadcast Service Providers

IRM (International Roaming MIN)
A form of MIN defined by IFAST (International Forum on ANSI-41 Standards Technology) towards facilitating international roaming by minimizing conflicts with the North American MIN.

Long-Term Key Message
Collection of keys and possibly, depending on the profile, other information like permissions and/or other attributes that are linked to items of content or services.

MIN (Mobile Identification Number)
MIN is a numeric ID that uniquely identifies a mobile defined by TIA standards for Cellular and PCS technologies. The MIN may be in the form of an IRM (International Roaming MIN). Note: the MIN may be in the form of the IRM.

Mobile Broadcast Service
Mobile Broadcast Services include a wide range of broadcast services, which jointly leverage both the unidirectional one-to-many broadcast paradigm and bi-directional unicast paradigm in a mobile environment, covering one-to-many services ranging from classical broadcast to mobile multicast. Typically, Mobile Broadcast Services deliver content suitable for simultaneous one-way distribution to a potentially large number of recipients without relying on specific addressing information of each recipient. Associated two-way interactive transactions having contextual relevance to the broadcast programs typically rely on established unicast delivery methods requiring specific recipient addressing information.

Examples of Mobile Broadcast Services include the following:

- pure Broadcast Services:
  - mobile TV
  - mobile newspaper
  - mobile file downloading
- combined broadcast/interactive Broadcast Services:
  - mobile TV for file downloading with voting
  - Broadcast Services for betting
  - Broadcast Services for auction
  - Broadcast Services for trading

Mobile Broadcast Service Provider
Business entity that has a role of providing the Mobile Broadcast Services to the user. Mobile Broadcast Service Provider may operate any set of server side functionalities as outlined in Mobile Broadcast Services Architecture [BCAST10-Architecture]. Mobile Broadcast Service Provider may have a subscription with the user. Note: In this specification Mobile Broadcast Service Provider is not technical or architectural concept

Mobility
The ability to receive service independent of location or while moving. (from OMA Dictionary)

Purchase Item
A purchase item groups one or multiple services or pieces of content that an end-user can purchase or subscribe to as a whole [BCAST10-SG].

Rights Issuer
An entity that issues Rights Objects to OMA DRM Conformant Devices [DRMDRM-v2.0].

Rights Object
A collection of Permissions, Constraints, and other attributes which define under what circumstances access is granted to, and what usages are defined for, DRM Content. All OMA DRM Conformant Devices must adhere to the Rights Object associated with DRM Content [DRMDRM-v2.0].

R-UIM
Acronym for 'Removable User Identity Module', corresponding to a non-UICC platform based module as defined in [3GPP2 C.S0023] to register services provided by 3GPP2 mobile networks with the appropriate security.

Short-Term Key Message
Message delivered alongside a protected service, carrying key material to decrypt and optionally authenticate the service, and access rights to delivered content.

Smartcard
A non-UICC secure function platform which may contain the SIM or R-UIM module, or a UICC-based secure function platform which may contain one or more of the following applications: a 3GPP USIM 3GPP2 CSIM. Note that the set of applications/modules residing on the Smartcard are typically governed

	by the affiliation of the Smartcard to 3GPP or 3GPP2 specifications, as indicated by the definition below for "Smartcard Profile".
Smartcard Profile	Alias for a set of Smartcard-based technologies and mechanisms which provide key establishment and key management, as well as permission and token handling for the Service and Content Protection solution for BCAST Terminals. In particular, subscriber key establishment and both short and long term key management may be based on GBA mechanisms and a Smartcard with (U)SIM as defined by 3GPP, or based on a pre-provisioned shared secret key and a Smartcard with R-UIM/CSIM or a UIM as defined by 3GPP2. The Smartcard Profile is described in [BCAST10-ServContProt] Section 6.
User ID	A unique ID that can be used to identify the user in the BCAST service areas of both the Home Mobile Broadcast Service Provider and the Visited Mobile Broadcast Service Provider. An example is the 3GPP/3GPP2 IMSI (International Mobile Subscriber Identity) as specified in 3GPP TS 23.003 and 3GPP2 C.S0005 (for the case the Broadcast Service Provider is a cellular mobile operator).
Visited Mobile Broadcast Service Provider	Any other Mobile Broadcast Service Provider than the user's Home Mobile Broadcast Service Provider.

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
BCAST	Mobile Broadcast Services
BCMCS	Broadcast Multicast Service
BDS	Broadcast Distribution System
BSA	BCAST Service Application
BSD/A	BCAST service distribution/adaptation
BSDA	BCAST Service Distribution and Adaptation
BSM	BCAST Subscription Management
BSM	BCAST Subscription Management
BSP-C	Broadcast service provisioning Client Function
BSP-M	Broadcast service provisioning Management Function
CID	Content ID
DCF	DRM Content Format
DRM	Digital Rights Management
DVB	Digital Video Broadcast
DVB-H	Digital Video Broadcast – Handheld
DVB-T	Digital Video Broadcast – Terrestrial
EN	European Norm
ESG	Electronic Service Guide
ETSI	European Telecommunications Standards Institute
FDT	File Delivery Table
FEC	Forward Error Correction
FLUTE	File Delivery over Unidirectional Transport
GZIP	GNU zip
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol

IC	Interaction Channel
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia Subsystem
INT	IP/MAC Notification Table
IP	Internet Protocol
IPDC	IP DataCast
IPsec	IP security
ISMACryp	Internet Streaming Media Alliance (ISMA) Encryption and Authentication
KMS	Key Management System
LTKM	Long-Term Key Message
MBMS	Multimedia Broadcast / Multicast Service
MIKEY	Multimedia Internet KEYing
MMS	Multimedia Messaging System
MPE	Multi-Protocol Encapsulation
MTD	Message Template Definition
OMA	Open Mobile Alliance
OSF	Open Security Framework
PSI/SI	Program Specific Information/Service Information
RI	Rights Issuer
RO	Rights Object
RTCP	Real Time Control Protocol
SDP	Session Description Protocol
SG	Service Guide
SG-C	Service Guide-Client
SG-D	Service Guide-Distribution
SGDU	Service Guide Delivery Unit
SIP	Session Initiation Protocol
SMIL	Synchronized Media Integration Language
SMS	Short Message Service
SRTP	Secure Real-time Transport Protocol
STKM	Short Term Key Message
TCP	Transmission Control Protocol
TP-C	Terminal Provisioning Client Component
TP-M	Terminal Provisioning Management Component
TR	Technical Report
TS	Technical Specification
UDP	User Datagram Protocol
WAP	Wireless Application Protocol
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

4. Introduction

The term "Mobile Broadcast Services" refers to a broad range of Broadcast Services, which jointly leverage the unidirectional one-to-many broadcast paradigm and the bi-directional unicast paradigm in a mobile environment, and covers one-to-many services ranging from classical broadcast to mobile multicast.

Building on mobile network systems, which provide bi-directional links, and digital broadcast systems, which provide uni-directional broadcast, Mobile Broadcast Services enable distribution of rich, interactive, and bandwidth consuming media content to large mobile audiences.

4.1 Version 1.0

In general, the availability of both broadcast channel and interaction channel are assumed for the BCAST 1.0 enabler. However, both broadcast channel and interaction channel may be temporarily unavailable, for example due to lack of radio coverage. Further, devices without access to an interaction channel are possible within the BCAST architecture and specifications. However, such devices may have limited functionality. Optimizations for devices without interaction channel are optional to implement in devices with interaction channel and are optional to use (for details see the SCR tables). Parts of the enabler are adaptation specifications for IPDC over DVB-H [BCAST10-DVBH-IPDC-Adaptation], 3GPP MBMS [BCAST10-MBMS-Adaptation], and 3GPP2 BCMCS [BCAST10-BCMCS-Adaptation].

This specification is structured as follows. Chapter 5 starts by mapping the interfaces as defined in BCAST Architecture [BCAST10-Architecture] to the various BCAST 1.0 Technical Specifications. Further, chapter 5 specifies the following BCAST 1.0 functions: Service Provisioning; Terminal Provisioning; Interaction, Personalization and Support for User-Based Profiles and Preferences; Charging; Mobility; Broadcast Roaming; Notification; and; Location Information. Appendix D provides informative examples related to service interaction and Appendix E illustrates the roaming related flows.

It is assumed that in BCAST 1.0 the network will make use of the BDS resources in accordance with the capabilities of the BDS.

5. Mobile Broadcast Services

Mobile Broadcast Services Architecture [BCAST10-Architecture] defines the Mobile Broadcast Services Enabler as a set of service-enabling functions. Within the overall architecture, each function has a set of interfaces, each of which forms the basis for interoperability. Although the architecture as such is not normatively specified, the interfaces provide a useful tool to map the various parts of BCAST specifications to the context of the overall architecture. The following table outlines how different parts of the BCAST Enabler are specified in the Technical Specifications.

Function	Interface	Normative Specification
Service Guide	SG-1	Out of scope of BCAST 1.0
	SG-2	Out of scope of BCAST 1.0
	SG-4	Refer to [BCAST10-SG], section 5.3 and 5.6
	SG-5	Refer to [BCAST10-SG], sections 5.3, 5.4.2 and 6.1.1
	SG-6	Refer to [BCAST10-SG], sections 5.3, 5.4.3, 6.1.2 and 6.2
	SG-B1	Refer to [BCAST10-SG], sections 5.3 and each BDS Adaptation Specification.
File Distribution	FD-1	Refer to [BCAST10-Distribution], section 5.4.1
	FD-2	Refer to [BCAST10-Distribution], section 5.4.1
	FD-5	Refer to [BCAST10-Distribution], section 5.2
	FD-6	Refer to [BCAST10-Distribution], section 5.3 and 5.5
	FD-B1	Refer to [BCAST10-Distribution] section 5.4.2 and each BDS Adaptation Specification.
Stream Distribution	SD-1	Refer to [BCAST10-Distribution], section 6.4.1
	SD-2	Refer to [BCAST10-Distribution], section 6.4.1
	SD-5	Refer to [BCAST10-Distribution], section 6.2
	SD-6	Refer to [BCAST10-Distribution], section 6.3 and 6.5
	SD-B1	Refer to [BCAST10-Distribution] section 6.4.2 and each BDS Adaptation Specification.
Service Protection	SP-2	Uses SD-2 and FD-2
	SP-4	Refer to [BCAST10-ServContProt] section 13.1
	SP-5-1	Refer to [BCAST10-ServContProt] section 5.6.1.1, 5.6.2.1, 6.8.1.1, and 6.8.2.1
	SP-5-2	Refer to [BCAST10-ServContProt] section 5.3, 5.4, 5.5, 6.5, 6.6, and 6.7
	SP-7	Refer to [BCAST10-ServContProt] section 5.3, 5.4, 6.5, and 6.6
	SP-9	Out of scope (this is a terminal internal interface and is not standardized within OMA BCAST)
Content Protection	CP-2	Uses SD-2 and FD-2
	CP-4	Refer to [BCAST10-ServContProt] section 13.2
	CP-5-1	Refer to [BCAST10-ServContProt] sections 5.6.1.2, 5.6.2.2, 6.8.1.2, and 6.8.2.2
	CP-5-2	Refer to [BCAST10-ServContProt] sections 5.3, 5.4, 5.5, 6.5, 6.6, and 6.7
	CP-7	Refer to [BCAST10-ServContProt] sections 5.3, 5.4, 6.5, and 6.6
	CP-9	Out of scope of BCAST 1.0 (this is a terminal internal interface and is not standardized within OMA BCAST)
Service Interaction	SI-8	Refer to this specification, section 5.3
Service Provisioning	SPR-7	Refer to this specification, section 5.1
	SPR-8	Out of scope (this interface is for out-of-band subscription)
Notification	NT-1	Refer to this specification, section 5.14

	NT-3	Refer to this specification, section 5.14
	NT-4	Refer to this specification, section 5.14
	NT-5	Refer to this specification, section 5.14
	NT-6	Refer to this specification, section 5.14
Terminal Provisioning	TP-4	Refer to this specification, section 5.2
	TP-5	Refer to this specification, section 5.2
	TP-7	Refer to this specification, section 5.2

Table 1: BCAST functions, Interfaces and Specifications

In addition to specific functions, the BCAST Enabler defines such horizontal, or universal, features as support for Mobility, Roaming and Charging. These aspects are in the scope of this specification.

5.1 Service Provisioning

BCAST Terminal SHALL support Service Provisioning messages if it supports the interaction channel and if it supports service and/or content protection as defined in [BCAST10-ServContProt]. This section specifies the messages used in Service Provisioning function over interface SPR-7, between Broadcast Service Provisioning Client (BSP-C) in the Terminal and Broadcast Service Provisioning Management (BSP-M) in the BSM. The Service Provisioning function supports the following operations:

- Requesting pricing information related to PurchaseItem declared in Service Guide
- Requesting / subscribing to service related to a PurchaseItem
- Renewing LTKMs related to already requested PurchaseItem
- Requesting /subscribing to a service that was already purchased (e.g. via out of band means)
- Cancelling a subscription related to already requested PurchaseItem
- Requesting a token or LTKM
- Inquiring the status of an account
- Subscription and unsubscription to user-specific notifications

To archive the above operations, the Service Provisioning function works with Service Guide function, Service Protection function, and Content Protection function. The linkage to Service Guide is through the use of PurchaseItem fragment which provides the identifiers (PurchaseItemID) used in the messages of Service Provisioning function. The linkage to Service and Content Protection function is through service request and subscription management messages, which requires the functionality of Service Protection Function and Content Protection Function.

This section has two sub-sections, one for BCAST general Service Provisioning message and one for Service Provisioning message based on Smartcard profile. BCAST General Provisioning messages supports the various kinds of Service Protection Function and Content Protection Function with the sub-elements and Smartcard service provisioning message are specified for Terminal supporting Smartcard profile.

The following two tables specify under which conditions each message is mandatory or optional to support for the general Service Provisioning message and Smartcard Service Provisioning message respectively.

Message	Section	Broadcast Service Provisioning Client (BSP-C)	Broadcast Service Provisioning Management(BSP-M)
Pricing Information Request	5.1.5.1.1	OPTIONAL	OPTIONAL
Pricing Information Response	5.1.5.1.2	MANDATORY	MANDATORY
Service Request	5.1.5.2.1	MANDATORY	MANDATORY
Service Response	5.1.5.2.2	MANDATORY	MANDATORY
Service Completion	5.1.5.2.3	OPTIONAL	MANDATORY
LTKM Renewal Request	5.1.5.3.1	MANDATORY	MANDATORY
LTKM Renewal Response	5.1.5.3.2	MANDATORY	MANDATORY
LTKM Renewal Completion	5.1.5.3.3	OPTIONAL	MANDATORY
Unsubscribe Request	5.1.5.4.1	MANDATORY	MANDATORY
Unsubscribe Response	5.1.5.4.2	MANDATORY	MANDATORY
Token Purchase Request	5.1.5.5.1	OPTIONAL	OPTIONAL
Token Purchase Response	5.1.5.5.2	OPTIONAL	OPTIONAL
Token Purchase Completion	5.1.5.5.3	OPTIONAL	OPTIONAL
Account Inquiry Request	5.1.5.6.1	MANDATORY	MANDATORY
Account Inquiry Response	5.1.5.6.2	MANDATORY	MANDATORY

Table 2: Summary General Service Provisioning messages

Message	Section	Broadcast Service Provisioning Client (BSP-C)	Broadcast Service Provisioning Management(BSP-M)
Pricing Information Request	5.1.6.1.1	OPTIONAL	OPTIONAL
Pricing Information Response	5.1.6.1.2	MANDATORY	MANDATORY
Service Request	5.1.6.2.1	MANDATORY	MANDATORY
Service Response	5.1.6.2.1	MANDATORY	MANDATORY
LTKM Renewal Request	5.1.6.3	MANDATORY	MANDATORY
LTKM Renewal Response	5.1.6.3	MANDATORY	MANDATORY
Unsubscribe Request	5.1.6.4.1	MANDATORY	MANDATORY
Unsubscribe Response	5.1.6.4.1	MANDATORY	MANDATORY
Token Purchase Request	5.1.6.5.1	MANDATORY	MANDATORY
Token Purchase Response	5.1.6.5.1	MANDATORY	MANDATORY
Token Purchase Completion	5.1.6.5.1	OPTIONAL	OPTIONAL
Account Inquiry Request	5.1.6.6.1	MANDATORY	MANDATORY
Account Inquiry Response	5.1.6.6.2	MANDATORY	MANDATORY
Registration Procedure	5.1.6.7	MANDATORY	MANDATORY
LTKM Request Procedure	5.1.6.8	MANDATORY	MANDATORY
Deregistration Procedure	5.1.6.9	MANDATORY	MANDATORY

Table 3: Summary Smartcard Service Provisioning messages

5.1.1 Transport Protocol for Service Provisioning Messages

Service Provisioning operations are executed by exchanging the Service Provisioning messages over interface SPR-7. All the Service Provisioning messages specified in the tables in the following sections and instantiated as XML documents.

All request and reply messages defined below contain a *requestID* field which MAY be used by a terminal to map a reply message to the corresponding request message. For this purpose, the network SHALL copy the requestID from a request message into to the corresponding reply message.

The URL towards which the service provisioning messages are directed is signaled through the PurchaseChannel fragment in SG as PurchaseURL [BCAST10-SG].

5.1.1.1 Transport Protocol for General Service Provisioning Messages

The BSP-M in the BSM SHALL support HTTP POST as a delivery method to exchange Service Provisioning messages over SPR-7.

The BSP-M in the BSM MAY support HTTPS POST as a delivery method to exchange Service Provisioning messages over SPR-7, where HTTPS SHALL be based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246].

The BSP-C in the Terminal SHALL support HTTP POST and MAY support HTTPS POST as a delivery method to exchange Service Provisioning messages over SPR-7, where HTTPS SHALL be based on .SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246].

For proper operation of Service Provisioning function, the terminal needs to know the URL for HTTP or HTTPS sessions. This is supported by ‘purchaseURL’ element contained in the PurchaseChannel fragment of Service Guide.

5.1.1.2 Transport Protocol for Smartcard Service Provisioning Messages

Most of the messages used for the Smartcard Profile are specified in [3GPP TS 33.246]. The remaining Service Provisioning messages are specified in the tables in the following sections and are instantiated as XML documents.

For the Smartcard Profile using (U)SIM or (R-)UIM/CSIM, the BSP-M in the BSM SHALL support HTTP POST and SHALL support HTTP digest authentication as per [3GPP TS 33.246] or [3GPP2 X.S0022-A], respectively, as a delivery method to exchange Service Provisioning messages over SPR-7.

For the Smartcard Profile using (U)SIM or (R-)UIM/CSIM, the BRP-C in the Terminal SHALL support HTTP POST and SHALL support HTTP digest authentication as per [3GPP TS 33.246] or [3GPP2 X.S0022-A], respectively.

For proper operation of Service Provisioning function, the terminal needs to know the URL for HTTP sessions. This is enabled by the ‘PurchaseURL’ element contained in the PurchaseChannel fragment of the Service Guide.

5.1.2 HTTP Binding

5.1.2.1 HTTP Binding for General Service Provisioning Message

Request messages are sent as HTTP content of type “application/vnd.oma.bcast.sprov+xml”. Responses are always sent as part of the “200 OK” response to the original request. The content type is “application/vnd.oma.bcast.sprov+xml”

5.1.2.2 HTTP Binding for Smartcard Service Provisioning Messages

HTTP Binding rule specified in [3GPP TS 33.246] SHALL be applied. If error is occurred on the procedure, HTTP response message SHALL have the error code defined in [3GPP TS 33.246]. If General Provisioning Messages are used, the same HTTP binding rule defined in the previous section will be applied.

5.1.3 Authentication

5.1.3.1 Message Authentication for General Service Provisioning Messages

For the general Service Provisioning messages, message authentication SHALL be provided using HTTPS that SHALL be based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246].

5.1.3.2 Subscriber Authentication for Smartcard Profile Service Provisioning Messages

Subscriber authentication for the Smartcard Profile SHALL be provided using HTTP digest as explained in [3GPP TS 33.246] or [3GPP2 X.S0022-A].

5.1.4 Use of Global Status Codes for Service Provisioning Messages

Table 4 proposes example values from Table 42 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 Table 42 are applicable to all messages if deemed required.

TS-BCAST_Services	5.1.5.1.2 Pricing Information Response	000, 001, 002, 003, 007, 008, 011, 013, 015, 016, 017, 018, 019, 020, 021, 023
	5.1.6.2.2 Service Response	000, 001, 002, 003, 004, 005, 006, 007, 008, 009, 011, 013, 014, 015, 016, 017, 018, 019 020, 021, 023, 031
	5.1.5.3.2 Long-Term Key Renewal Response	000, 001, 002, 004, 005, 006, 007, 008, 010, 011, 013, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024,
	5.1.5.4.2 Unsubscribe Response	000, 001, 002, 007, 008, 010, 011, 013, 015, 016, 017, 018, 019, 020, 021, 022, 023
	5.1.5.5.2 Token Purchase Response	000, 001, 002, 004, 005, 006, 007, 008, 009, 011, 013, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024
	5.1.5.6.2 Account Inquiry Response	000, 001, 002, 004, 005, 007, 008, 011, 013, 014, 015, 017, 018, 019, 020, 021, 023
	5.7.2.3. Roaming Authorization Response	000, 001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 013, 014, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024, 025, 026
	5.7.2.5 RoamingServiceResponse	000, 001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 013, 014, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024, 025, 026

Table 4: Cross Reference Table (Informative)

5.1.5 General Service Provisioning Messages

This section specifies the General Service Provisioning Messages. As described, many of the messages in this category support the Service Provisioning function of both the Smartcard Profile and DRM Profile BCAST Terminals, whereas others specifically pertain to Service Provisioning for DRM Profile terminals. The XML schema for these messages is defined in [BCAST10-XMLSchema-orderqueries].

5.1.5.1 Pricing Information Request Messages

This message is sent by the terminal to the BSM to request the pricing information of a particular purchase item or items. It is used in the following situations:

- the Service Guide announces Purchase Data elements associated with the Purchase Item, but does not announce any price for some or all of them, or
- the user wishes to discover whether a different price or additional purchase options are available for his or her subscriber ID.

The response message returns information about the price and subscription options for each purchase item, and optionally the full Service Guide fragments that describe them.

In the case of the (U)SIM Smartcard Profile, the terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

5.1.5.1.1 Pricing Information Request

Name	Type	Category	Cardinality	Description	Data Type
PricingInfo Request	E			Pricing Information Request Message. Contains the following attributes: requestID Contains the following elements: UserID DeviceID PurchaseItem BroadcastRoamingSpecificPart	
requestID	A	O	0..1	Identifier for the Price Information request message.	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, this element SHALL be included. For the Smartcard profile, this element SHALL be omitted, and the user identity SHALL be provided by the network with HTTP DIGEST authentication procedure defined in section 6.6 Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile, this element SHALL be included if the device supports IMEI or MEID. A device supporting the DRM profile SHALL NOT allow the user to modify the DeviceID. Contains the following attributes: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 — reserved for future use 1 – IMEI [3GPP TS 23.003] 2 – MEID [3GPP2 C.S0072]	unsignedByte

				3-127 reserved for future use 128-255 reserved for proprietary use	
Purchase Item	E1	M	1..N	Identifier of the Purchase Item for which the user wants to know the price. Contains the following attribute: globalIDRef	
globalIDRef	A	M	1	The ID of the Purchase Item. A purchase item is identified by the GlobalPurchaseItemID found in the PurchaseItem fragment.	anyURI
PurchaseDataReference	E2	O	0..N	Identifier the PurchaseData fragments for which the user wishes to know the price. If this element is omitted, the user is asking for the price of all the Purchase Data fragments associated with the Purchase Item, and available to the particular user.	
idRef	A	M	1	Identification of the 'PurchaseData' fragment in question.	anyURI
BroadcastRoamingSpecificPart	E1	O	0..1	This element provides information to help processing the Service Request in case of roaming. For rules on how to use this element, see section 5.7.3. If the BSM support Broadcast Roaming, it SHALL support this element. If the Terminal support Broadcast Roaming, it SHALL support this element. Contains the following elements: HomeBSM VisitedBSM	
HomeBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Visited BSM, this element indicates the Home BSM of the terminal in the context of this request.	complexType as defined for 'BSMFilter Code' in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Home BSM, this element indicates the Visited BSM from which the user wishes to purchase service.	complexType as defined for 'BSMFilter Code' in section 5.4.1.5.2 of [BCAST10-SG]

Table 5: Structure of Pricing Information Request in General Service Provisioning Message

5.1.5.1.2 Pricing Information Response

If the price information request is accepted by BSM, then the message from BSM contains following data:

Name	Type	Category	Cardinality	Description	Data Type

PricingInfoResponse	E			Pricing Information Response Contains the following attributes: requestID globalStatusCode Contains the following elements: PurchaseItem	
requestID	A	O	0..1	Identifier for the corresponding Pricing Information request message or Service Request message.	unsignedInt
global Status Code	A	M	0..1	The overall outcome of the request, according to the return codes defined in the section 5.11. ▪ If this attribute is present and set to value “0”, the request was completed successfully. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘PurchaseItem’. ▪ If this attribute is present and set to some other value than “0”, there was a generic error concerning the entire request. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘PurchaseItem’. ▪ If this attribute is not present, there was an error concerning one or more ‘PurchaseItem’ elements associated with the request. Further, the ‘itemwiseStatusCode’ SHALL be given per each requested ‘PurchaseItem’.	unsignedByte
PurchaseItem	E1	M	0..N	Describes the purchase-related information of a purchase item requested in the related PricingInfoRequest message or ServiceRequest message. It is possible to provide one or more prices of a purchase item by currency. This element SHALL not be instantiated in case the ‘globalStatusCode’ attribute is present and set to a value different from ‘0’. In any other case, it SHALL be instantiated. In case the child ‘itemwiseStatusCode’ indicates success, or the ‘globalStatusCode’ is present and set to ‘0’, at least one of ‘PurchaseDataReference’ or ‘PurchaseDataFragment’ element SHALL be instantiated. Note that it is permitted to include instances of both ‘PurchaseDataReference’ and ‘PurchaseDataFragment’ elements into the same response. Contains the following attribute: globalIDRef	

				itemwiseStatusCode Contains the following element: PurchaseDataReference PurchaseDataFragment	
globalIDRef	A	M	1	Identifier of the Purchase Item for which a price was requested. A purchase item is identified by the GlobalPurchaseItemID found in the PurchaseItem fragment.	anyURI
itemwise Status Code	A	M	0..1	Specifies a status code of each PurchaseItems using GlobalStatusCode defined in the section 5.11.	unsignedByte
PurchaseData Reference	E2	O	0..N	Describes the purchase-related options available for this user. Contains the following attribute: idRef Contains the following elements: Price SubscriptionPeriod SubscriptionType TermsOfUse	
idRef	A	M	1	Identifier of this Purchase Data, to be used by the terminal when referencing to the purchase data in a subsequent Service Request message.	anyURI
Price	E3	M	1..N	Price information of purchase item that a user wants to know. This element takes precedence over the 'MonetaryPrice' element of the referenced PurchaseData fragment. Contains the following attributes: validTo currency	decimal
validTo	A	O	0..1	The last moment when this price information is valid. If not given, the validity is assumed to end in undefined time in the future. This field expressed as the first 32bits integer part of NTP time stamps. The validity indicated by this attribute SHALL be equal to or be within the range of the fragment validity of the associated 'PurchaseData' fragment.	unsignedInt
currency	A	M	1	Specifies the currency codes defined in ISO 4217 international currency codes.	string
SubscriptionPeriod	E3	O	0..1	Specifies the subscription period for the option represented by this PurchaseData. If the Purchase Item represents a bundle of services, the SubscriptionPeriod SHALL be returned. Otherwise it MAY be omitted. This element takes precedence over the 'SubscriptionPeriod'	duration

				element of the referenced PurchaseData fragment.	
SubscriptionType	E3	M	1	The type of subscription offered as defined in section 5.1.2.7 of [BCAST10-SG]. Allowed values are: 0 – one-time subscription 1 – open-ended subscription 2 – free trial subscription 3 – (not applicable) 4 – 127 Reserved for future use 128-255 Reserved for proprietary use The Token-based modes defined in the PurchaseData fragment SHALL NOT be signalled here.	unsignedByte
TermsOfUse	E3	O	0..N	Element that declares there are Terms of Use associated with the 'PurchaseData' fragment and parent 'PurchaseItem' this 'Pricing Information Response' relates to. 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. Multiple occurrences of 'TermsOfUse' are allowed within this message, but for any two such occurrences values for elements "Country" and "Language" SHALL NOT be same at the same time. Contains the following attributes: type id userConsentRequired Contains the following sub-elements: Country Language PreviewDataIDRef TermsOfUseText	
type	A	M	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 message. 1 – Not used. 2 - 127 reserved for future use	unsignedByte

				128 -255 reserved for proprietary use	
id	A	M	1	The URI uniquely identifying the Terms of Use.	anyURI
userConsentRequired	A	M	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 message. false: User consent is not required for the Terms of Use.	boolean
Country	E4	O	0..N	List of countries for which the Terms of Use is 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 of three digits
Language	E4	M	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
PreviewDataIDRef	E4	O	0..1	Reference to the PreviewData fragment which carries the representation of legal text. If this element is not present, the 'TermsOfUseText' element SHALL be present (Implementation in XML schema using <choice>).	anyURI
TermsOfUseText	E4	O	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
PurchaseDataFragment	E2	O	0..N	This element holds PurchaseData fragments in the format specified in [BCAST10-SG] This element SHALL NOT be used to provide a PurchaseData fragment that does not relate to a purchase item requested by the user	Complex Type

Table 6: Structure of Pricing Information Response in General Service Provisioning Message

5.1.5.2 Service Request Message

This message is sent by the terminal to the BSM to request the subscription to, or purchase of, the associated purchase item(s), and is applicable to both the DRM Profile and Smartcard Profile. This message is used strictly for the subscription/purchase of purchase item(s) which is(are) not associated with token-based payment. The Smartcard Profile also uses this message to submit a request for a SEK/PEK associated with a specific Key Validity period (range of STKM Time Stamp values), when the SEK/PEK required to enable play-back of protected recording is not available on the Smartcard (see Section 6.9.1 of [BCAST10-ServContProt]).

In the case of the (U)SIM Smartcard Profile, the terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

5.1.5.2.1 Service Request

This message is sent by the terminal to the BSM to request the subscription to, or purchase of, the associated purchase item.

If the price is specified in the request message and it differs from the price calculated by the BSM for one or more of the purchase items included in the request, the BSM SHALL respond with Pricing Information Response message (5.1.5.1.2).

Also, if the price is not specified for one or more of the purchase items in the request message, the BSM SHALL respond with Pricing Information Response message (5.1.5.1.2). Otherwise, the BSM SHALL respond with Service Response message (5.1.5.2.2).

In a similar fashion, in case the ServiceRequest message does not contain an instance of the 'UserConsentAnswer' element for a 'PurchaseItem' element, while it is expected that the user agrees to terms of use at the time of subscription for the said 'PurchaseItem', the BSM MAY respond with a PricingInformationResponse message that contains the 'TermsOfUse' element for the PurchaseItem(s) requiring it, or return the error code '31' in the itemwiseStatusCode indicating that BSM rejected the subscription because the user did not agree to the terms of use. In the latter case the terminal MAY issue a PrincingInformationRequest to obtain the terms of use.

In case the BSM answers a Service Request with a Pricing Information Response, the latter SHALL list at least all those purchase items requested in the related Service Request for which subscription-related information (e.g. pricing, terms of use, subscription type) is absent or incorrect. The terminal SHALL consider it has accurate subscription-related information for those purchase items provided in the Service Request but not present in the Princing Information Response.

Name	Type	Category	Cardinality	Description	Data Type
ServiceRequest	E			Service Request Message to subscribe or purchase PurchaseItem Contains the following attributes: requestID Contains the following elements: UserID DeviceID ServiceEncryptionProtocol PurchaseItem DrmProfileSpecificPart BroadcastRoamingSpecificPart The Service Request message MAY contain an instance of the DrmProfileSpecificPart element.	
requestID	A	O	0..1	Identifier for the Service request message.	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, this element SHALL be included. For the Smartcard profile, this element SHALL be omitted, and the user identity SHALL be provided by the network with HTTP DIGEST authentication procedure defined in section 6.6 Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865]	unsignedByte

				1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	
DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile this element SHALL be included if the device supports IMEI or MEID. A device supporting the DRM profile. SHALL NOT allow the user to modify the DeviceID. Contains the following attributes: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – reserved for future use 1 – IMEI [3GPP TS 23.003] 2 – MEID [3GPP2 C.S0072] 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
ServiceEncryptionProtocol	E1	O	0..N	Lists each service encryption protocol supported by the device, including the mandatory ones. Defined values: “ipsec”, “srtp”, and “ISMACryp”. The device is allowed to include more identifiers, however depending on the protocols supported by the network they may be ignored. Note: This element is only included in the message if a service is to be delivered over Interaction channel.	string
Purchase Item	E1	M	1..N	Contains the list and price of items the user wants to order and the list of services the user wants to subscribe notification. Contains the following attributes: globalIDRef Contains the following elements: PurchaseDataReference Service	
globalIDRef	A	M	1	The identifier of the Purchase Item. The Purchase Item identifier is advertised in the PurchaseItem fragment of the Service Guide as GlobalPurchaseItemID and is inserted in this message in the same format.	anyURI
PurchaseDataReference	E2	O	0..1	Contains the price information. This specifies the PurchaseData fragment in the Service Guide which is to be used for this subscription. Contains the following attribute idRef Contains the following Element:	

				Price	
idRef	A	M	1	References the identifiers of PurchaseData Fragment advertised in Service Guide.	anyURI
Price	E3	O	0..1	The price of the Purchase Item known to the user from Service Guide. If PurchaseData in the Service Guide contains multiple price entries by currency, this element should be specified to indicate to the BSM the entry desired by the user. Contains the following attribute: currency	decimal
currency	A	O	0..1	Specifies the currency codes defined in ISO 4217 international currency codes.	string
UserConsentAnswer	E2	O	0..1	Signals whether user agreed to the Terms of Use as represented by id of the related TermsOfUse element. true: User agrees the terms of the Terms of Use. false: User disagrees the terms of the Terms of Use. If this element is not present the interpretation is that the user has not read or understood the Terms of Use.	boolean
id	A	M	1	The URI uniquely identifying the Terms of Use this 'UserConsentAnswer' relates to, which is declared either in a PurchaseData fragment, or a PurchaseChannel fragment. Said otherwise, the 'UserConsentAnswer' parent element relates to Terms of Use applicable to a PurchaseData-PurchaseItem pair.	anyURI
Service	E2	O	0..N	Reference of the Service. This element is only used for subscribing service-specific Notification. As of this version of the specification, it is assumed that service-specific Notifications delivered over the Broadcast Channel do not require subscription as they are sent in the clear. Hence, this element only applies for subscription to service-specific Notification delivered over the Interaction Channel. Contains the following attributes: globalIDRef notification Note: This element is only used for the purpose of subscribing to service-specific Notification. In addition, this element should not be confused with the MBMS User Service ID (the latter is the equivalent MBMS designation for the concatenation of the attributes 'PurchaseItemID.@gobalIDRef' and 'PurchaseData.@idRef' in BCAST.	
globalIDRef	A	M	1	Unique ID of the Service, as represented by the GlobalServiceID of the 'Service' fragment. It is	anyURI

				used to identify the Service to which the service-specific Notification relates..	
notification	A	M	1	This attribute declares whether subscription to receive service-specific Notification message over the Interaction Channel is required. If set to ”true”, the terminal wishes to subscribe to delivery of the service-specific Notification over the Interaction Channel. If set to ”false”, the terminal does not wish to subscribe to delivery of service-specific Notification over Interaction Channel.	boolean
DrmProfile SpecificPart	E1	O	0..1	Service & Content Protection DRM-profile specific part. This part is MANDATORY to support for the DRM Profile, and is not applicable to the Smartcard Profile. Contains the following attributes: rightsIssuerURI Contains the following element: BroadcastMode	
rightsIssuer URI	A	O	0..1	ID of the rights issuer associated with the BSM.	anyURI
Broadcast Mode	E2	O	0..1	Indicates whether or not the device supports the optional broadcast mode of operation for rights acquisition, in addition to the interactive mode of operation.	boolean
BroadcastRoamingSpecificPart	E1	O	0..1	This element provides information to help processing the Service Request in case of roaming. For rules on how to use this element, see section 5.7.3. If the BSM support Broadcast Roaming, it SHALL support this element. If the Terminal support Broadcast Roaming, it SHALL support this element.	
HomeBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Visited BSM, this element indicates the Home BSM of the terminal in the context of this request.	complexType as defined for ‘BSMFilter Code’ in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Home BSM, this element indicates the Visited BSM from which the user wishes to purchase service.	complexType as defined for ‘BSMFilter Code’ in section 5.4.1.5.2 of [BCAST10-SG]

Table 7: Structure of Service Request in General Service Provisioning Message

5.1.5.2.2 Service Response

This message is sent to the terminal from the BSM in response to the request for subscription to the Service Request message. This message is applicable to both the DRM Profile and Smartcard Profile.

Name	Type	Category	Cardinality	Description	Data Type
ServiceResponse	E			Service Response Message Contains the following attributes: requestID globalStatusCode adaptationMode Contains the following elements: PurchaseItem DrmProfileSpecificPart SmartcardProfileSpecificPart	
requestID	A	O	0..1	Identifier for the corresponding Service request message.	unsignedInt
global Status Code	A	M	0..1	The overall outcome of the request, according to the return codes defined in section 5.11. This attribute also governs the way the 'itemwiseStatusCode' attribute is instantiated in this response: ▪ If this attribute is present and set to value "0", the request was completed successfully. In this case the 'itemwiseStatusCode' SHALL NOT be given per each requested 'PurchaseItem'. ▪ If this attribute is present and set to some other value than "0", there was a generic error concerning the entire request. In this case the 'itemwiseStatusCode' SHALL NOT be given per each requested 'PurchaseItem'. ▪ If this attribute is not present, there was an error concerning one or more 'PurchaseItem' elements associated with the request. Further, the 'itemwiseStatusCode' SHALL be given per each requested 'PurchaseItem'.	unsignedByte
adaptation Mode	A	O	0..1	Informs the terminal of the operational adaptation mode: Generic or BDS-specific adaptation false – indicates Generic adaptation mode true – indicates BDS-specific adaptation mode Note: this attribute SHALL be present only if the 'globalStatusCode' indicates "Success", and the underlying BDS is BCMCS.	boolean
PurchaseItem	E1	M	0..N	Describes the results of the request message of subscribing to or purchasing the PurchaseItem. For the DRM Profile, if subscription or	

				purchase is successful, rightsValidityEndTime of PurchaseItem will be present. For either the DRM Profile or Smartcard Profile, in the case of subscription/purchase failure, itemwiseStatusCode MAY be present to indicate the reason why the request is not accepted by BSM. This element SHALL NOT be instantiated in case the ‘globalStatusCode’ attribute is present and set to a value different from ‘0’. In any other condition of the ‘globalStatusCode’ attribute, it SHALL be instantiated. Contains the following attributes: globalDRef itemwiseStatusCode Contains the following element: SubscriptionWindow	
globalIDRef	A	M	1	The ID of the Purchase Item. A purchase item is identified by the GlobalPurchaseItemID found in the PurchaseItem fragment.	anyURI
itemwiseStatusCode	A	M	0..1	Specifies a status code of each PurchaseItems using GlobalStatusCode defined in the section 5.11.	unsignedByte
SubscriptionWindow	E2	O	0..1	The time interval during which the subscription is valid. The network SHOULD include this element for time-based subscriptions and MAY include it for pay-per-view. The terminal MAY use this information to determine the validity period of a subscription. Contains the following attributes: startTime endTime	
startTime	A	M	1	NTP timestamp expressing the start of subscription.	unsignedInt
endTime	A	O	0..1	NTP timestamp expressing the end of subscription. This attribute SHALL NOT be present for open-ended subscriptions.	unsignedInt
DrmProfile SpecificPart	E1	O	0..1	Service & Content Protection DRM-profile specific part. This part is MANDATORY to support for the DRM Profile, and is not applicable to the Smartcard Profile. This element SHALL NOT be instantiated in case the ‘globalStatusCode’ attribute is present and set to a value different from ‘0’. In any other case, it MAY be instantiated. Contains the following attributes: rightsValidityEndTime Contains the following elements:	

				roap Trigger	
rights Validity EndTime	A	O	0..1	The last time and date of validity of the Long-Term Key Message, after which it has to be renewed. This attribute will be present when BSM accept the request message. This field is expressed as the first 32bits integer part of NTP time stamps. Note: this element is validated if RO is broadcasted. Otherwise, this element is not necessary.	unsignedInt
roap Trigger	E2	O	0..1	ROAP RO Acquisition Trigger**. The device is expected to use the trigger to initiate one or more Long-Term Key Message acquisitions.	reference to "roapTrigger" element as defined in OMA DRM 2.0 XML namespace
SmartcardProfileSpecificPart	E1	O	0..1	Service & Content Protection Smartcard-profile specific part. This part is MANDATORY to support for the Smartcard Profile, and is not applicable to the DRM Profile. Contains the following elements: LTKM	
LTKM	E2	O	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message). This element MAY be present - if the terminal and the BSM have agreed on "HTTP" as a LTKM delivery mechanism during the registration procedure (see section 5.1.6.10), and the BSM wishes to deliver LTKM to the terminal at the moment the ServiceResponse is done.	base64Binary

Table 8: Structure of Service Response in General Service Provisioning Message

** These (ROAP Messages) are DRM profile specific. They are defined in [DRMDRM-v2.0].

5.1.5.2.3 Service Completion (DRM Profile only)

This message MAY be sent by a terminal after it has received a Service Response Message and retrieved all LTKMs. The network SHALL reply with a HTTP 200 OK response message when this message is received.

Name	Type	Category	Cardinality	Description	Data Type
ServiceCompletion	E			Service Completion Message Message. Contains the following attribute: requestID Contains the following element: LTKMessageID	
requestID	A	O	0..1	Identifier for the corresponding Service request	unsignedInt

				message.	
LTK MessageID	E1	M	1..N	A list containing the IDs of one or more LTKMs received by the device. This is the RO ID .	string

Table 9: Structure of Service Completion in General Service Provisioning Message

5.1.5.3 LTKM Renewal Messages

The following messages in this section are specific to the DRM Profile. For the Smartcard Profile, the equivalent messages and procedures pertaining to LTKM renewal are defined in Section 5.1.6.3.

5.1.5.3.1 LTKM Renewal Request (DRM Profile only)

The Long-term Key Message Renewal request message is sent if a terminal needs to renew the LTKM(s) associated to a certain Purchase Item or group of purchase items. It is only applicable to the DRM Profile.

This message can also be sent by the terminal to the BSM to request the subscription to any purchase items that the end user has already purchased (e.g. via out of band means), but has not yet received key material for. This could for example be used the first time the BCAST application is started in order to register the terminal to “free” or “default” channels.

Name	Type	Category	Cardinality	Description	Data Type
LTKMRenewalRequest	E			Long Term Key Message Renewal Request Message Contains the following attributes: requestID Contains the following elements: UserID DeviceID PurchaseItem	
requestID	A	O	0..1	Identifier for the LTKM renewal request message.	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, this element SHALL be included. Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile, this element SHALL be	string

				included if the device supports IMEI or MEID. A device supporting the DRM profile SHALL NOT allow the user to modify the DeviceID Contains the following attributes: type	
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – reserved for future use 1 – IMEI [3GPP TS 23.003] 2 – MEID [3GPP2 C.S0072] 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
Purchase Item	E1	M	1..N	A list of Purchase Items that the user wants to renew. Contains the following attribute: globalIDRef If the terminal wants to request from the BSM the delivery of a list of all purchase items that the end user has already purchased, the terminal has to set the globalIDRef attribute equal to "**oma-bcast-allservices**". This could for example be used the first time the BCAST application is started in order to register the terminal to "free" or "default" channels. If the terminal wants to request from the BSM a list of all those purchase items that the end user has already purchased (e.g. via out of band means), but has not yet received a ROAP trigger for, the terminal has to set the globalIDRef attribute equal to "**oma-bcast-newservices**". If either "**oma-bcast-allservices**" or "**oma-bcast-newservices**" is used, there SHALL be exactly one 'PurchaseItem element' in the request.	
globalIDRef	A	M	1	GlobalPurchaseItemID to identify this PurchaseItem, found in the PurchaseItem fragment.	anyURI

Table 10: Structure of LTKM renewal request in General Service Provisioning Message

5.1.5.3.2 LTKM Renewal Response (DRM Profile only)

Name	Type	Category	Cardinality	Description	Data Type
LTKMRenewalResponse	E			Long Term Key Message Renewal Response Message Contains the following attributes: requestID globalStatusCode Contains the following elements:	

				PurchaseItem DrmProfileSpecificPart	
requestID	A	O	0..1	Identifier for the corresponding LTKM request message.	unsignedInt
global Status Code	A	M	0..1	▪ The overall outcome of the request, according to the return codes defined in section 5.11.If this attribute is present and set to value “0”, the request was completed successfully. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘PurchaseItem’. ▪ If this attribute is present and set to some other value than “0”, there was a generic error concerning the entire request. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘PurchaseItem’. ▪ If this attribute is not present, the request was completed successfully but there was an error concerning one or more ‘PurchaseItem’ elements associated with the request. Further, the ‘itemwiseStatusCode’ SHALL be given per each requested ‘PurchaseItem’. In case this message is a response to an LTKMRenewalRequest with ‘globalIDRef’ set to “**oma-bcast-newservices**” or “**oma-bcast-allservices**”, an empty result list SHALL be signalled by setting ‘globalStatusCode’ equal to “010” (No Subscription) and not instantiating the ‘PurchaseItem’ element.	unsignedByte
PurchaseItem	E1	M	0..N	Describes the results of the request message of LTKM Renewal. If renewal is successful, LTKValidityEndTime of PurchaseItem will be present. If not, itemwiseStatusCode will be present to show user the reason why the request is not accepted by BSM. This element SHALL NOT be instantiated in case the ‘globalStatusCode’ attribute is present and set to a value different from ‘0’. In any other case, it SHALL be instantiated. Contains the following attributes: globalIDRef ltkValidityEndTime itemwiseStatusCode Contains the following sub-elements: SubscriptionWindow PurchaseDataReference In case the globalIDRef attribute of the	

				PurchaseItem element has been set equal to "**oma-bcast-allservices**" in the corresponding request message, the reply message SHALL contain a list of all PurchaseItem elements which the terminal has already purchased and which it is entitled to access currently or in the future In case the globalIDRef attribute of the PurchaseItem element has been set equal to "**oma-bcast-newservices**" in the corresponding request message, the reply message SHALL contain a list of those PurchaseItem elements which the terminal has already purchased (e.g. via out of band means) and which it is entitled to access currently or in the future, but for which it has not received key material.	
globalIDRef	A	M	1	The ID of the Purchase Item to which the validity end time is related. A purchase item is identified by the GlobalPurchaseItemID found in the PurchaseItem fragment.	anyURI
ltkValidityEndTime	A	O	0..1	The last time and date of validity of the Long-Term Key Message, after which it has to be renewed again. This attribute will be present when BSM accept the request message. This field is expressed as the first 32bits integer part of NTP time stamps. Note: the information on this element can be provided in RO.	unsignedInt
itemwiseStatusCode	A	M	0..1	Specifies a status code of each PurchaseItems using GlobalStatusCode defined in the section 5.11.	unsignedByte
SubscriptionWindow	E2	O	0..1	The time interval during which the subscription is valid. The server MAY omit this element if the response is not successful. Otherwise: • For time-based subscriptions, the network SHALL include this element when responding to an "**oma-bcast-allservices**" or "**oma-bcast-newservices**" request and SHOULD include it otherwise. • For pay-per-view, the network MAY include this element. The terminal MAY use this information to determine the validity period of a subscription. Contains the following attributes: startTime endTime	
startTime	A	M	1	NTP timestamp expressing the start of subscription.	unsignedInt

endTime	A	O	0..1	NTP timestamp expressing the end of subscription. This attribute SHALL NOT be present for open-ended subscriptions.	unsignedInt
PurchaseDataReference	E2	O	0..1	Describes the PurchaseData associated with the subscription to the Purchase. The device MAY use this information to update its internal subscription information concerning the user. The server SHALL include this element if the response is successful, and MAY omit it if it is not. Contains the following attributes: idRef Contains the following sub-element: Price	
idRef	A	M	1	The id of the Purchase Data fragment that is being referred to.	anyURI
Price	E3	O	0..N	The price currently associated for the use to the subscription, possibly in multiple currencies. Contains the following attribute: currency	decimal
currency	A	O	0..1	Specifies the currency codes defined in ISO 4217 international currency codes. If not given, value of price is amount of Tokens.	string
DrmProfile SpecificPart	E1	O	0..1	Service & Content Protection DRM-profile specific part. This part is MANDATORY to support for the DRM Profile. Note that as this message is only applicable for the DRM profile, this element SHALL always be present for successful responses (i.e. 'globalStatusCode' being equal to 0 or not instantiated). Contains the following elements: Trigger	
Trigger	E2	O	0..1	ROAP RO Acquisition Trigger**. If the LTKM renewal failed because the device was unregistered, the response MAY include a ROAP Registration Trigger**. In that case, the device is expected to use the trigger to initiate a registration and repeat the LTKM renewal once it is registered.	RoapTrigger

Table 11: Structure of LTKM renewal response in General Service Provisioning Message

** These (ROAP Messages) are DRM profile specific

5.1.5.3.3 LTKM Renewal Completion (DRM Profile Only)

This message MAY be sent by the terminal to the BSM as an acknowledgment of the terminal's receipt of the LTKM Renewal Response and subsequent retrieval of all related LTKMs. The network SHALL reply with a HTTP 200 OK response message when this message is received.

Name	Type	Category	Cardinality	Description	Data Type
LTKMRenewalCompletion	E			Long-Term Key Message Renewal Completion Message	

				Contains the following attributes: requestID Contains the following elements: LongTermKeyID	
requestID	A	O	0..1	Identifier for the corresponding LTKM request message.	unsignedInt
LongTerm KeyID	E1	M	1..N	A list containing the IDs of one or more Long-Term Key Messages received by the device.	string

Table 12: LTKM renewal completion in General Service Provisioning Message

5.1.5.4 Unsubscription Messages

These messages pertain to the request and response for cancellation of the existing subscription to the purchase item as identified by the 'globalIDRef attribute' of PurchaseItem or the notification as identified by the 'globalIDRef attribute' of Service.

Depending on the specific situation, a subscription could still be valid after this procedure has been successfully executed, for example because the user has already paid a non-refundable amount for a time span that is yet to elapse. In this case, the subscription is to be considered valid until the time indicated in the "subscribedUntil" attribute of the response.

A device supporting the DRM Profile SHALL continue to renew keys with the LTK renewal procedure while the subscription is still valid, even if the user has unsubscribed.

When the device unsubscribing supports the smartcard profile, some additional actions need to occur upon successful completion of the unsubscribe procedure. The BSM MAY also invalidate SEKs associated with the relevant purchase ID on the unsubscribing device which are not used by any other purchase items to which the device is subscribed. The BSM invalidates SEKs/PEKs by sending an LTKM with invalid Key Validity data, i.e. the lower bound is greater than the upper bound, where the bounds define the allowed range of either TEK IDs or TimeStamp values.

In the case of the (U)SIM Smartcard Profile, the terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

5.1.5.4.1 Unsubscribe Request

Name	Type	Category	Cardinality	Description	Data Type
UnsubscribeRequest	E			Unsubscribe Request Message Contains the following attributes: requestID keepSubscription Contains the following elements: UserID DeviceID PurchaseItem	
requestID	A	O	0..1	Identifier for the Unsubscribe request message.	unsignedInt
keepSubscription	A	O	0..1	This element declares whether this UnsubscribeRequest message requests un-subscription from both the PurchaseItem and related service-specific Notification delivered over the Interaction Channel, or only the latter. When the user wants to unsubscribe from	boolean

				service-specific Notifications delivered over the Interaction Channel but keep the subscription to PurchaseItem, this attribute SHALL be set to "true". If this attribute is not present or holds value "false", it means both PurchaseItem and its relevant notification are requested for un-subscription.	
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, this element SHALL be included. For the Smartcard profile, this element SHALL be omitted, and the user identity SHALL be provided by the network with HTTP DIGEST authentication procedure defined in section 6.6. Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile this element SHALL be included if the device supports IMEI or MEID. A device supporting the DRM profile SHALL NOT allow the user to modify the DeviceID. Note: If a user has multiple devices, then this element indicates a device or a group of devices that the user wants to unsubscribe. Contains the following attribute: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – reserved for future use 1 – IMEI [3GPP TS 23.003] 2 – MEID [3GPP2 C.S0072] 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
Purchase Item	E1	M	1..N	Specifies identifier of the Purchase Item the user wants to unsubscribe from. Also, contains ServiceID to unsubscribe service-specific notification. Contains the following attribute: globalIDRef	

				Contains the following element: Service	
globalIDRef	A	M	1	Identifier of PurchaseItem. GlobalPurchaseItemID found in the PurchaseItem fragment will be used.	anyURI
Service	E2	O	0..N	This element is only used for unsubscribing service-specific Notification. See section 5.14.4.2.1. As of this version of the specification, it is assumed that service-specific Notifications delivered over the Broadcast Channel do not require un-subscription as they are sent in the clear. Hence, this element only applies for un-subscription from service-specific Notification delivered over the Interaction Channel. Contains the following attributes: globalIDRef notification	
globalIDRef	A	M	1	GlobalServiceID of the 'Service' fragment to identifying the service to which this service-specific Notification relates.	anyURI
notification	A	M	1	This attribute declares un-subscription from delivery of the service-related Notification over the Interaction Channel is required. If set to "true", the terminal wishes to unsubscribe from delivery of service specific Notification over the Interaction Channel. If set "false" or is absent, it means there is no change in current status of subscription for service-specific Notification delivered over the Interaction Channel.	boolean

Table 13: Structure of Unsubscribe Request in General Service Provisioning Message

5.1.5.4.2 Unsubscribe Response

Name	Type	Category	Cardinality	Description	Data Type
UnsubscribeResponse	E			Unsubscribe Response Message Contains the following attributes: requestID globalStatusCode Contains the following elements: PurchaseItem	
requestID	A	O	0..1	Identifier for the corresponding Unsubscribe request message.	unsignedInt
global Status Code	A	M	0..1	The overall outcome of the request, according to the return codes defined in section 5.11. ▪ If this attribute is present and set to value "0", the request was completed successfully. In this case the 'itemwiseStatusCode' SHALL NOT be given per each requested 'PurchaseItem'.	unsignedByte

				▪ If this attribute is present and set to some other value than "0", there was a generic error concerning the entire request. In this case the 'itemwiseStatusCode' SHALL NOT be given per each requested 'PurchaseItem'. ▪ If this attribute is not present, there was an error concerning one or more 'PurchaseItem' elements associated with the request. Further, the 'itemwiseStatusCode' SHALL be given per each requested 'PurchaseItem'.	
Purchase Item	E1	M	1..N	The ID of the Purchase Item to which the message is related. This element SHALL NOT be instantiated in case the 'globalStatusCode' attribute is present and set to a value different from '0'. In any other case, it SHALL be instantiated. Contains the following attribute: globalIDRef itemwiseStatusCode	
globalIDRef	A	M	1	Identifier of PurchaseItem. GlobalPurchaseItemID found in the PurchaseItem fragment will be used.	anyURI
itemwiseStatus Code	A	M	0..1	Indicates the results of the Unsubscribe Request message. If Value is successful, it means relevant PurchaseItem is unsubscribed. GlobalStatusCode specified in section 5.11 will be used for this code.	UnsignedBy te
subscribedUnti l	A	O	0..1	The date and time until which the subscription is still valid. If missing, the subscription is to be considered terminated immediately. For the DRM profile, this is the time until which the terminal SHALL continue to issue LTK renewal requests for the purchase item. For Smartcard Profile, this is the time until which the BSM SHALL continue to include the purchase item in subsequent registration responses. This field is expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
SmartcardProf ileSpecificPart	E1	O	0..1	Service & Content Protection Smartcard-profile specific part. This part is MANDATORY to support for the Smartcard Profile, and is not applicable to the DRM Profile. Contains the following elements: LTKM	
LTKM	E2	O	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message). This element is present if the terminal and the BSM have agreed on "HTTP" as a LTKM delivery	base64Binar y

				mechanism during the registration procedure (see section 5.1.6.10)	

Table 14: Structure of Unsubscribe Response in General Service Provisioning Message

5.1.5.5 Token Purchase Request Messages

5.1.5.5.1 Token Purchase Request

This message is sent by the terminal to the BSM to request the purchase of tokens, or credits, to enable future consumption of broadcast services/content. The quantity of which is identified by the requested token amount. This message is applicable to both the DRM Profile and Smartcard Profile.

In the case of the (U)SIM Smartcard Profile, the terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

Name	Type	Category	Cardinality	Description	Data Type
TokenPurchaseRequest	E			Token Purchase Request Message Contains the following attributes: requestID Contains the following elements: UserID DeviceID PermissionsIssuerURI TokensRequested BroadcastRoamingSpecificPart	
requestID	A	O	0..1	Identifier for the Token Purchase request message.	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, this element SHALL be included. For the Smartcard profile, this element SHALL be omitted, and the user identity SHALL be provided by the network with HTTP DIGEST authentication procedure defined in section 6.6 Contains the following attribute: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte

DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile this element SHALL be included if the device supports IMEI or MEID. A device supporting the DRM profile SHALL NOT allow the user to modify the DeviceID. Contains the following attribute: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – reserved for future use 1 – IMEI 3GPP TS 23.003 2 – MEID 3GPP2 C.S0072 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
PermissionsIssuerURI	E1	O	0..1	The identification of the Permissions Issuer depending on the Profile. For the DRM Profile, this element is MANDATORY. It identifies the Rights Issuer from which the BSM can retrieve the ROAP Trigger**. For the Smartcard Profile, this element SHALL NOT be instantiated as only the BSM can grant tokens in the case of the Smartcard Profile. Contains the following attribute: type	anyURI
type	A	M	1	The type of the Permissions Issuer identified by the PermissionsIssuerURI. Allowed values are: false – DRM Profile true – Reserved for future use As of this version of the specification, this attribute SHALL be set to “false” when instantiated.	boolean
TokensRequested	E1	O	0..1	Purchase request for tokens Contains the following attributes: type amount chargingType purchaseUnitNum Contains the following elements: PurchaseItem SmartCardProfileSpecificPart	
type	A	M	1	Specifies the type of tokens requested Allowed values are: 0 - unspecified 1 – tokens for the DRM Profile 2 – service tokens for the Smartcard Profile, added to the live_ppt_purse of the specified SEK/PEK key group 3 – service tokens for the Smartcard Profile, to the playback_ppt_purse of the specified	unsignedByte

				SEK/PEK key group 4 – user tokens for the Smartcard Profile added to the user purse associated to the BSM ID 5 - 127 reserved for future use 128-255 reserved for proprietary use Note: type 1 tokens are applicable only to DRM Profile, whereas types 2-4 are applicable only to Smartcard Profile For a definition of user tokens and service tokens, see Sections 6.6.4.2 and 6.6.7 of [BCAST10-ServContProt].	
amount	A	M	1	For types 0 and 1, this value corresponds to the number of tokens requested in this Token Purchase Request message. For types 2 and 3, this value corresponds to the number of service tokens contained in a single service token-based credit package. These tokens are valid for any LTKM using service tokens associated to the given SEK/PEK key group. For type 4, this value corresponds to the requested number of user tokens, valid for any LTKM using user tokens associated to the ID of the BSM.	unsignedInt
charging Type	A	O	0..1	The type of charging (pre-paid or post-paid) the user wishes to use. The BSM will verify that the requested charging type is available for this user. The following values are defined: 0 – undefined 1 – prepaid 2 – postpaid 3-127 – reserved for future use 128-255 – reserved for proprietary use If this attribute is not present, the default value is 0.	unsignedByte
purchaseUnit Num	A	O	0..1	The number of token-based credit packages requested by the terminal, where the number of tokens in one package is indicated by ‘amount’ attribute above. If this field is absent, then the request is for one package only (i.e. the default value is 1.) The value of the ‘amount’ attribute SHALL be identical to the value of ‘TotalNumberCredits’ element specified in the associated ‘PurchaseData’ fragment in the SG. Therefore the actual number of tokens requested by the terminal is ‘purchaseUnitNum’ times ‘amount’. Note that ‘PurchaseUnitNum’ SHOULD be limited in accordance to the Purchase Data fragment associated with the Purchase Item of concern in the SG. For example, in the case of play-based tokens, its maximum value SHOULD equal that of the attribute	unsignedShort

				‘maxReplay’ under ‘TotalNumberTokenCredits’, assuming the attribute ‘extraTokensPurchaseable’ of ‘CreditPackageType’ has value = 1.	
PurchaseItem	E2	O	0..1	Identifier of the purchase item to which the type of tokens in the token purchase request corresponds, if the information comes from the Service Guide and the request relates to a PurchaseItem. This is given by the globalPurchaseItemID as defined in [BCAST10-SG]. Contains the following attributes: globalIDRef purchaseDataIDRef For Smartcard profile this field MAY be present if the request is for user tokens and MAY be present if the request is for service tokens. This field MAY be absent if the request is for DRM profile tokens.	
globalIDRef	A	M	1	Identifier of PurchaseItem. GlobalPurchaseItemID found in the PurchaseItem fragment will be used.	anyURI
purchaseDataIDRef	A	O	0..1	Identifies the associated ‘PurchaseData’ fragment to which the requested credit package belongs.	anyURI
SmartcardProfileSpecificPart	E2	O	0..1	Service & Content Protection Smartcard Profile specific part. This part is MANDATORY to support for the Smartcard Profile, and is not applicable to the DRM Profile. Contains the following elements: ProtectionKeyID	
ProtectionKeyID	E3	M	0..1	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 specified in the Smartcard Profile [BCAST10-ServContProt]. The ProtectionKeyID corresponds to the SEK/PEK ID for which service tokens are requested. The element is only present when service tokens are requested AND the PurchaseItem element is absent. When user tokens are requested, ‘ProtectionKeyID’ SHOULD be absent, since the received user tokens in a subsequent LTKM are deposited into the user purse.	base64Binary
BroadcastRoamingSpecificPart	E1	O	0..1	This element provides information to help processing the Service Request in case of roaming. For rules on how to use this element, see section 5.7.3. If the BSM support Broadcast Roaming, it SHALL support this element. If the Terminal support Broadcast Roaming, it	

				SHALL support this element.	
HomeBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Visited BSM, this element indicates the Home BSM of the terminal in the context of this request.	complexType as defined for 'BSMFilter Code' in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSM	E2	M	0..1	In case the Service Provisioning request is issued against the Home BSM, this element indicates the Visited BSM from which the user wishes to purchase service.	complexType as defined for 'BSMFilter Code' in section 5.4.1.5.2 of [BCAST10-SG]

Table 15: Structure of Token Purchase Request in General Service Provisioning Message

** These (ROAP Messages) are DRM profile specific

5.1.5.5.2 Token Purchase Response

This message, sent from the BSM to the terminal, represents a successful outcome, either unconditional or conditional in nature, in response to the Token Purchase Request. This message is applicable to both the DRM Profile and Smartcard Profile.

Name	Type	Category	Cardinality	Description	Data Type
TokenPurchaseResponse	E			Token Purchase Response Contains the following attributes: requestID globalStatusCode Contains the following elements: TokensGranted DrmProfileSpecificPart SmartcardProfileSpecificPart Note: DrmProfileSpecificPart and SmartcardProfileSpecificPart are mutually exclusive – TokenPurchaseResponse SHALL contain either the DrmProfileSpecificPart or SmartcardProfileSpecificPart.	
requestID	A	O	0..1	Identifier for the corresponding Token Purchase request message.	unsignedInt
globalStatusCode	A	M	1	The outcome of the request, according to the return codes defined in section 5.11.	unsignedByte
TokensGranted	E1	O	0..1	Granted tokens in response to the token purchase request. It contains the following attributes: type	

				amount chargingType Note: The element TokensGranted simply represents the information on the outcome of the token purchase request. The actual token delivery is fulfilled by a LTKM.	
type	A	M	1	Specifies the type of tokens granted in the token purchase transaction. Allowed values are: 0 – reserved 1- tokens for DRM Profile 2 – service tokens for the Smartcard Profile, added to the live_ppt_purse of the specified SEK/PEK key group 3 – service tokens for the Smartcard Profile added to the playback_ppt_purse purse of the specified SEK/PEK key group 4 – user tokens for the Smartcard Profile added to the user purse associated to the BSM ID 5-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
amount	A	M	1	Specifies the number of tokens granted in the token purchase transaction. For type 0, 1, 2, 3 and 4, the value corresponds to the number of tokens granted. Note that this value is not equal to the attribute ‘amount’ given in the Token Purchase Request message. In the Token Purchase Response, ‘amount’ represents the total number of tokens sought by the terminal in the associated token purchase request, i.e. it equals the product (amount) x (PurchaseUnitNum) in that request.	unsignedInt
charging Type	A	O	0..1	The type of charging to be associated with the token purchase transaction. The following values are defined: 0 – unspecified 1 – prepaid 2 – postpaid 3-127 – reserved for future use 128-255 – reserved for proprietary use If this attribute is not present, the default value is 0.	unsignedByte
DrmProfileSpecificPart	E1	O	0..1	Service & Content Protection DRM-profile specific part. This part is MANDATORY to support for the DRM Profile, and is not applicable to the Smartcard Profile.. Contains the following elements: roap Trigger	
roap Trigger	E2	O	0..1	If the token purchase succeeded, the response SHALL include a ROAP Trigger** as an	reference to “roapTrigge

				additional payload. The device is expected to use the trigger to initiate one or more token acquisitions. If the token purchase failed because the device was unregistered, the response includes a ROAP Registration Trigger** as an additional payload. The device is expected to use the trigger to initiate a registration and repeat the token purchase once it is successfully registered.	r" element as defined in OMA DRM 2.0 XML namespace
SmartcardProfileSpecificPart	E1	O	0..1	Service & Content Protection Smartcard Profile specific part. This part is MANDATORY to support for the Smartcard Profile, and is not applicable to the DRM Profile. Contains the following element: LTKM	
LTKM	E2	O	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message). This element is present if the terminal and the BSM have agreed on "HTTP" as a LTKM delivery mechanism during the registration procedure (see section 5.1.6.10)	base64Binary

Table 16: Structure of Token Purchase Response in General Service Provisioning Message

***These (ROAP messages) are OMA DRMv2.0 specific. They are defined in [DRMDRM-v2.0]. Implementation in XML schema will be done by referenceing the "RoapTrigger element from the OMA DRM2.0 ROAP protocol schema. Other service protection mechanisms will map their own respective messages to the corresponding fields.*

5.1.5.5.3 Token Purchase Completion

Token Purchase Completion Message MAY be sent by a terminal after it receives Token Purchase Response Message.

Name	Type	Category	Cardinality	Description	Data Type
TokenPurchaseCompletion	E			Token Purchase Completion Message for terminal to send. Contains the following attributes: requestID	
requestID	A	O	0..1	Identifier for the corresponding Token Purchase request message.	unsignedInt

Table 17: Structure of Token Purchase Completion in General Service Provisioning Message

5.1.5.6 Account Inquiry Messages

Account Inquiry allows the user to request his/her account information such as active PurchaseItem list, associated PurchaseData and Billing Information. The AccountInquiry Element in the Account Inquiry Request message (5.1.5.6.1) indicates which information the user wants to receive and the response message can include billing information or a list of purchase items, possibly complemented by purchase data and the related fragments, as requested by the value of the 'type' attribute in the request message.

In the case of the (U)SIM Smartcard Profile, the terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

5.1.5.6.1 Account Inquiry Request

Name	Type	Category	Cardinality	Description	Data Type
AccountRe	E			Account Inquiry Request message	

quest				Contains the following attributes: requestID Contains the following elements: UserID DeviceID AccountInquiry	
requestID	A	O	0..1	Identifier for this request message	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. For the DRM profile, element SHALL be included if the device supports IMEI or MEID. For the Smartcard profile, this element SHALL be omitted, and the user identity SHALL be provided by the network with HTTP DIGEST authentication procedure defined in section 6.6 Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
DeviceID	E1	O	0..N	A unique device identification known to the BSM. For the DRM profile this element SHALL be included if the device supports IMEI or MEID. A device supporting the DRM profile SHALL NOT allow the user to modify the DeviceID. Contains the following attribute: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – DVB Device ID 1 – 3GPP Device ID (IMEI) [3GPP TS 23.003] 2 – 3GPP2 Device ID (MEID) [3GPP2 C.S0072] 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
AccountInquiry	E1	M	1..N	Specifies the account information which user want to receive from the BSM. Possible values are: 0 – undefined 1 – PurchaseItem list 2 – PurchaseItem list with a copy of the	unsignedByte

				applicable PurchaseItem fragments 3 – Billing Information 4 – PurchaseItem and PurchaseData list 5– PurchaseItem and PurchaseData list with a copy of the applicable fragments 6 ~ 127 – Reserved for future use 128 ~ 255 – Reserved for proprietary use If value is 0, BSM SHOULD deliver the response message as either one of the message types defined above, or as defined by the BSM operator. If value is ‘1’, the BSM SHOULD respond with one or more instances of the ‘PurchaseItem’ element. There MAY be no instance of this element in the response in case there is no applicable purchase item. The ‘BillingInformation’, ‘PurchaseItemFragment’ and ‘PurchaseData’ elements SHALL NOT be instantiated. If value is ‘2’, the BSM SHOULD respond as for value ‘1’ and MAY additionally provide an instance of the ‘PurchaseItemFragment’ element under the ‘PurchaseItem’ element. The ‘BillingInformation’ and ‘PurchaseData’ elements SHALL NOT be instantiated. If value is ‘3’, the BSM SHOULD instantiate the ‘BillingInformation’ element in the response. There MAY be no instance of this element in the response in case there is no applicable billing information. The ‘PurchaseItem’ element SHALL NOT be instantiated. If value is ‘4’, the BSM SHOULD respond as for value ‘1’ and additionally instantiate the ‘PurchaseData’ element with an ‘idRef’ attribute in the response. There MAY be no instance of the 'PurchaseData' element in the response in case there is no applicable purchase information. The ‘BillingInformation’ , ‘PurchaseItemFragment’ and ‘PurchaseDataFragment’ elements SHALL NOT be instantiated. If value is ‘5’, the BSM SHOULD respond as for value ‘4’ and in addition MAY provide an instance of the ‘PurchaseItemFragment’ element under the ‘PurchaseItem’ element and MAY provide an instance of the ‘PurchaseDataFragment’ element under the	

				‘PurchaseData’ element. The ‘BillingInformation’ element SHALL NOT be instantiated.	

Table 18: Structure of Account Inquiry Request in General Service Provisioning Message

5.1.5.6.2 Account Inquiry Response

Name	Type	Category	Cardinality	Description	Data Type
AccountResponse	E			Account Inquiry Response Message Contains the following attributes: requestID globalStatusCode Contains the following elements: BillingInformation PurchaseItem	
requestID	A	O	0..1	Identifier for the corresponding Account Inquiry message	unsignedInt
global Status Code	A	M	1	The overall outcome of the request, according to the return codes defined in section 5.11.	unsignedByte
BillingInformation	E1	O	0..N	Describes the total billing information, possibly in multiple languages. The language is expressed using built-in XML attribute xml:lang with this element.	string
PurchaseItem	E1	O	0..N	Specifies a PurchaseItem to which the user subscribed or purchased. Contains the following attributes: globalIDRef Contains the following elements: Description PurchaseItemFragment PurchaseData	
globalIDRef	A	M	1	GlobalPurchaseItemID of Purchase Item which the End user subscribed or purchased.	anyURI
Description	E2	O	0..N	Describes the subscription information such as price, period, etc., possibly in multiple languages. The language is expressed using built-in XML attribute xml:lang with this element.	string
PurchaseItemFragment	E2	O	0..1	Contains the PurchaseItem Fragment related to the PurchaseItem to which the End user subscribed or purchased.	complexType as defined for ‘PurchaseItem’ in section 5.1.2.6 of [BCAST10-SG]
PurchaseData	E2	O	0..1	Specifies the PurchaseData fragment applicable	

				to the PurchaseItem to which the user subscribed or purchased.	
idRef	A	M	1	Identifier of the PurchaseData fragment	anyURI
PurchaseDataFragment	E3	O	0..1	Contains a copy of the PurchaseData fragment declared by the parent 'PurchaseData' fragment.	complexType as defined for 'PurchaseData' in section 5.1.2.7 of [BCAST 10-SG]

Table 19: Structure of Account Inquiry Response in General Service Provisioning Message

5.1.6 Smartcard Profile Service Provisioning Messages

This section specifies the Smartcard Service Provisioning Messages. These messages support the Service Provisioning function of BCAST Terminals with Smartcard Profile capability. The messages in Sections 5.1.6.1, 5.1.6.2 and 5.1.6.4 through 5.1.6.6 below are identical to General Service Provisioning Messages. The messages in Section 5.1.6.3 are somewhat unique as described in the corresponding section below. The messages in Sections 5.1.6.7 through 5.1.6.9 are unique to the Smartcard Profile (i.e. no counterparts for these exist in the General Service Provisioning Messages).

The XML schema for these messages is defined in [BCAST10-XMLSchema-orderqueries].

5.1.6.1 Pricing Information Messages

5.1.6.1.1 Pricing Information Request

This message is the same as the general service provisioning message. See section 5.1.5.1.1.

5.1.6.1.2 Pricing Information Response

This message is the same as the general service provisioning message. See section 5.1.5.1.2.

5.1.6.2 Service Request Messages

Service Request and Service Response messages are the same as those specified in Section 5.1.5.2.

Although there is no Service Completion message for the Smartcard profile, the BSM can determine if the terminal has successfully received the Service Response by requesting an LTKM verification message (specified in section 6.6.6.1 of [BCAST10-ServContProt]) in at least one of the LTKMs subsequently transmitted to the terminal in the context of this Service Request procedure. The LTKM verification message(s) sent by BCAST terminal to BSM will confirm to BSM the successful reception of this Service Response.

5.1.6.3 LKTM Renewal, Response and Completion Messages

LTKMs can be explicitly renewed with a Registration Procedure (Section 5.1.6.7), the LTKM Request Procedure (Section 5.1.6.8) or implicitly renewed via MSK delivery procedure as described in [3GPP TS 33.246].

Although there is no LTKM Renewal Completion message for the Smartcard profile, the BSM can determine if the terminal has successfully received the LTKM(s) by requesting an LTKM verification message (specified in section 6.6.6.1 of [BCAST10-ServContProt]) in the LTKM(s) transmitted to the terminal in the context of this LTKM Renewal procedure. The LTKM verification message(s) sent by BCAST terminal to BSM will confirm to BSM the successful reception of the LTKM(s).

5.1.6.4 Unsubscription Messages

5.1.6.4.1 Unsubscribe Request and Response

These messages are the same as those specified in Section 5.1.5.4.

5.1.6.5 Token Messages

These messages are the same as those specified in Section 5.1.5.5.

5.1.6.6 Account Inquiry Messages

These messages are the same as the General Service Provisioning Account Inquiry messages as specified in Section 5.1.5.6.

This message is the same as the general service provisioning message. See section 5.1.5.6.2

5.1.6.7 Registration Procedure

The Registration procedure is invoked by the terminal when the BCAST Client is started or re-activated and upon re-establishing connectivity to the interactivity network after having lost coverage or in response to a BSM Solicited Pull Procedure where BM-SC Solicited Pull message is formatted according to-Section 6.6.2 of [BCAST10-ServContProt].

In order to ensure proper LTKM delivery mechanism negotiation prior to LTKM delivery, the terminal SHALL besides perform a Registration procedure:

- before a terminal-initiated LTKM Request procedure, if terminal has not yet registered to the PermissionsIssuerURI of the Access fragment describing service access.
- before a Service Provisioning procedure subject to LTKM delivery (Service Request, Token Purchase, Unsubscription), if terminal has not yet registered to the PermissionsIssuerURI of the Access fragment (indirectly) associated with the purchase item.

The Registration procedure is used by the terminal to notify the BSM that it is available to receive LTKMs or parental control messages. The Registration procedure is not used in OMA BCAST to request any change in the subscription/ purchase status of the terminal. This functionality is provided by the Service Provisioning messages, e.g. Service Request.

For the (U)SIM Smartcard Profile terminal, this procedure is the MBMS User Service Registration procedure as defined by [3GPP TS 33.246], in which one single MBMS User Service ID is indicated in the Registration Request: the value is “oma-bcast-allservices”. The Registration Response returned by the BSM SHALL indicate the total list of (PurchaseItem, PurchaseData) for which the terminal is authorized to receive the related LTKMs (including LTKMs to invalidate SEKs/PEKs). More specifically, the response SHALL contain one Response element per MBMS User Service ID. Each MBMS User Service ID identifies a (PurchaseItem, PurchaseData) pair, encoded as the concatenation of GlobalPurchaseItemID and PurchaseDataReference values. Items that are unsubscribed but still valid due to the presence of the “subscribedUntil” attribute in the “Unsubscribe Response” message SHALL be also included in the Registration Response. In case there are no such items available to return, there SHALL be exactly one “Response” element with “serviceID” set to the reserved identifier “oma-bcast-noservices” and “ResponseCode” set to “200 OK”. In this particular registered state, the BSM SHALL NOT send LTKMs, BSM solicited pull procedure initiation messages and Parental Control messages to the terminal, but MAY send BSM solicited pull messages to trigger re-registrations.

The above procedure is not applicable in the case of the (R-)UIM/CSIM Smartcard Profile, i.e., when BCMCS is the underlying BDS.

The terminal SHALL handle the PDP context used for this procedure as specified in section 5.1.6.12.

The terminal MAY include in the registration request one RegistrationRequestExtension in order to:

- indicate the LTKM delivery mechanisms it supports starting from the time of this request. This mechanism is defined in sections 5.1.6.7.1 and 5.1.6.10.1 and 5.1.6.11.1.

- indicate the minimal intended lifetime of terminal PDP context after the completion of MBMS-based and Service Provisioning HTTP procedures. This mechanism is defined in sections 5.1.6.7.1 and 5.1.6.12.

The BSM MAY include in the registration response one RegistrationResponseExtension in order to:

- indicate the LTKM delivery mechanisms it plans to use for further messages deliveries to the terminal. This mechanism is defined in sections 5.1.6.7.2, 5.1.6.10.1 and 5.1.6.11.1.

- specify the minimal lifetime of terminal PDP context after the completion of MBMS-based and Service Provisioning HTTP procedures. This mechanism is defined in sections 5.1.6.7.2 and 5.1.6.12.

The BSM can also include in the registration response one or several RegistrationResponseServiceExtensions in order to:

- deliver the LTKMs the terminal is authorized to receive and any parental control messages. This information MAY be included. The underlying mechanism is defined in sections 5.1.6.7.2, 5.1.6.10.3 and 5.1.6.11.2.

- indicate the subscription start and end times of the PurchaseItem/PurchaseData pairs for which the terminal is authorized to receive the related LTKMs. For time-based subscriptions, this information SHALL be. For pay-per-view, this information MAY be included.

The following is an informative example illustrating the BCAST extensions (printed in boldface) possibly present in a Registration Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<mbmsSecurityRegisterResponse
    xmlns="urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:bcast="urn:oma:xml:bcast:pr:orderqueries:1.0">
    <Response>
        <serviceID>urn:3gpp:mbms:example:service:identification:123456789abcdef</serviceID>
        <ResponseCode>200 OK</ResponseCode>
        <bcast:RegistrationResponseServiceExtension>
            <LTKM>...</LTKM>
            <SubscriptionWindow startTime="3408134400" endTime=”3410812800”/>
        </bcast:RegistrationResponseServiceExtension version="0">
    </Response>
    <Response>
        <serviceID>urn:3gpp:mbms:example:service:identification:fedcba987654321</serviceID>
        <ResponseCode>200 OK</ResponseCode>
        <bcast:RegistrationResponseServiceExtension>
            <LTKM>...</LTKM>
            <LTKM>...</LTKM>
            <SubscriptionWindow startTime="3408134400" endTime=”3410812800”/>
        </bcast:RegistrationResponseServiceExtension>
    </Response>
   <bcast:RegistrationResponseExtension version="0">
       <LTKMDelivery>
           <Trigger>1</Trigger> <!-- indicates ‘SMS’ -->
           <Type>1</Type>      <!-- indicates ‘HTTP’ -->
       <LTKMDelivery>
       <PDPContextLifetime>0</PDPContextLifetime>
   </bcast:RegistrationResponseExtension>
</mbmsSecurityRegisterResponse>
```

5.1.6.7.1 Registration Request Extension

The Registration Request payload is an “mbmsSecurityRegister” message defined according to XML schema “urn:3GPP: metadata:2005:MBMS:securityRegistrationRequest” specified in section 11.4.1 of [3GPP TS 26.346 v7].

To allow the inclusion of BCAST-specific information at <mbmsSecurityRegister> level of Registration Request payload, a *RegistrationRequestExtension* element is defined in the namespace “urn:oma:xml:bcast:pr:orderqueries:1.0” [BCAST10-XMLSchema-orderqueries]. When included, this element SHALL be present exactly once, as a child of <mbmsSecurityRegister> element matching the <xs:any> wildcard defined there.

The *RegistrationRequestExtension* element is structured as follows:

Name	Type	Category	Cardinality	Description	Data Type
RegistrationRequestExtension	E		0..1	Defines a container for the inclusion of BCAST-specific information at the <mbmsSecurityRegister> level of Registration Request payload defined in section 11.4.1 of [3GPP TS 26.346 v7]. Contains the following attributes: version Contains the following elements: LTKMDelivery PDPContextLifetime	
version	A	NM/ TM	1	Version of this extension element. 0x00 identifies BCAST 1.0	unsignedByte
LTKMDelivery	E1	NO/ TO	0..1	This element lists all the LTKM and parental control message delivery mechanisms the terminal will support from this registration request till next registration request. Detailed use of this element is further specified in section 5.1.6.10.1and 5.1.6.11.1. Contains the following elements: Trigger Type	
Trigger	E2	NM/TM	1..N	Specifies the trigger delivery mechanisms supported by the terminal at the time of registration request (triggers designating: messages to initiate BSM solicited pull procedure, and BSM solicited pull messages to initiate registration). Allowed values are: 0 – UDP 1 – SMS as per section 5.1.6.10.2 2 – 127 reserved for future use 128 – 255 reserved for proprietary use	unsignedByte
Type	E2	NM/ TM	1..N	Specifies the LTKM and parental control message delivery mechanisms supported by the terminal at the time of registration request. Allowed values are: 0 – UDP 1 – HTTP as per section 5.1.6.10.3 and 5.1.6.11.2 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
PDPContext Lifetime	E1	NO/ TO	0..1	Approximate minimal lifetime in seconds during which terminal intends to maintain its PDP context alive, after the completion of each MBMS-based procedure (registration, deregistration, LTKM request) and Service	unsignedInt

				Provisioning procedure (Service Request, Token Purchase, Unsubscription), whether these procedures are initiated by terminal or triggered by BSM. Note: how the terminal can determine an appropriate PDP context lifetime value for the interaction network in use is out of the scope of this specification This number of seconds SHALL be counted starting from the approximate time of sending/reception of the last message of the HTTP procedure (Registration Response, Service Response, etc). The terminal SHOULD NOT set this number of seconds to zero, as the BSM may not support other means than UDP for the delivery of LTKMs, triggers and Parental Control messages. When the element is absent, the lifetime spans till the completion of next De-registration procedure wih this BSM. The use of this element is further specified in section 5.1.6.12.	

Table 20: Structure of RegistrationRequestExtension

5.1.6.7.2 Registration Response Extension

The Registration Response payload is an “mbmsSecurityRegisterResponse” message defined according to XML schema “urn:3GPP: metadata:2005:MBMS:securityRegistrationResponse” specified in section 11.7.1 of [3GPP TS 26.346 v7].

To allow the inclusion of BCAST-specific information at <mbmsSecurityRegisterResponse> level of Registration Response payload, a *RegistrationResponseExtension* element is defined in the namespace “urn:oma:xml:bcast:pr:orderqueries:1.0” [BCAST10-XMLSchema-orderqueries]. When included, this element SHALL be present once as a child of <mbmsSecurityRegisterResponse> element matching the <xs:any> wildcard defined there.

This *RegistrationResponseExtension* element is structured as follows:

Name	Type	Category	Cardinality	Description	Data Type
RegistrationResponseExtension	E		0..1	Defines a container for the inclusion of BCAST-specific information at the <mbmsSecurityRegisterResponse> level of Registration Response payload defined in section 11.7.1 of [3GPP TS 26.346 v7]. Contains the following attributes: version Contains the following sub-elements: LTKMDelivery ParentalControlMessage PDPContextLifetime	
version	A	NM/ TM	1	Version of this extension element. 0x00 identifies BCAST 1.0	unsignedByte

LTKMDelivery	E1	NO/ TO	0..1	This element lists all the LTKM and parental control message delivery mechanisms the BSM plans to use from this registration response (included) till next terminal registration request occurs. Detailed use of this element is further specified in section 5.1.6.10.1 and 5.1.6.11.1. Contains the following elements: Trigger Type	
Trigger	E2	NM/TM	1..N	Specifies the delivery mechanisms which the BSM intends to use to deliver triggers to the terminal till next registration request (triggers designating: messages to initiate BSM solicited pull procedure, and BSM solicited pull messages to initiate registration). Allowed values are: 0 – UDP 1 – SMS as per section 5.1.6.10.2 2 – 127 reserved for future use 128 – 255 reserved for proprietary use	unsignedByte
Type	E2	NM/ TM	1..N	Specifies the delivery mechanisms which the BSM intends to use to deliver LTKMs and parental control messages to the terminal, till next registration request. Allowed values are: 0 – UDP 1 – HTTP as per section 5.1.6.10.3 and 5.1.6.11.2 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
ParentalControlMessage	E1	NO/TO	0..1	Smartcard profile BCAST Parental control message (base64-encoded MIKEY message) as defined in section 6.6.5 of [BCAST10-ServContProt]. This element is used to deliver parental control messages via HTTP, in case the terminal and the BSM have agreed on “HTTP” as a delivery mechanism for LTKM during this registration procedure (see section 5.1.6.10) This element SHALL be supported in case HTTP delivery of LTKMs and Parental control messages is supported.	base64Binary
PDPContext Lifetime	E1	NO/ TO	0..1	Approximate minimal lifetime in seconds during which terminal SHALL maintain its PDP context alive, after the completion of each MBMS-based procedure (registration, deregistration, LTKM request) and Service Provisioning procedure (Service Request, Token Purchase, Unsubscription), whether these procedures are initiated by terminal or triggered by BSM. This number of seconds SHALL be counted starting from the approximate time of	unsignedInt

				sending/reception of the last message of the HTTP procedure (Registration Response, Service Response, etc). The BSM MAY set this number of seconds to zero, when UDP bearer is not a negotiated mechanism for the delivery of LTKMs, triggers and Parental Control messages When the terminal indicates in Registration Request a PDP context lifetime in seconds greater than zero, the BSM SHOULD replicate this number of seconds in the returned <PDPContextLifetime> element. When the element is not included by the BSM, the lifetime spans till completion of De-registration with this BSM. The use of this element is further specified in section 5.1.6.12.	

Table 21: Structure of RegistrationResponseExtension

5.1.6.7.3 Registration Response Service Extension

The Registration Response payload is an “mbmsSecurityRegisterResponse” message defined according to XML schema “urn:3GPP: metadata:2005:MBMS:securityRegistrationResponse” specified in section 11.7.1 of [3GPP TS 26.346 v7].

To allow the inclusion of BCAST-specific information at <Response> level of Registration Response payload (i.e. at the level corresponding to one registered PurchaseItem/PurchaseData pair), a *RegistrationResponseServiceExtension* element is defined in the namespace “urn:oma:xml:bcast:pr:orderqueries:1.0” [BCAST10-XMLSchema-orderqueries]. This element MAY be included in each/any <Response> element in the Registration Response. When included in a <Response> element, it SHALL be present once as a child of <Response> element matching the <xs:any> wildcard defined there.

This *RegistrationResponseServiceExtension* element is defined below:

Name	Type	Category	Cardinality	Description	Data Type
RegistrationResponseServiceExtension	E		0..1	Defines a container for the inclusion of BCAST-specific information at the <Response> level of Registration Response payload defined in section 11.7.1 of [3GPP TS 26.346 v7]. Contains the following attributes: version Contains the following elements: LTKM	
version	A	NM/ TM	1	Version of this extension element. 0x00 identifies BCAST 1.0	unsignedByte
LTKM	E1	NO/ TO	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message) associated with the successfully registered PurchaseItem/PurchaseData pair identified by <serviceID> element sibling of <RegistrationResponseServiceExtension> element.	base64Binary

				This element SHALL NOT be included if <ResponseCode> element sibling of <RegistrationResponseServiceExtension> does not indicate status code "200 OK". More details on this element are further specified in section 5.1.6.10.3.	
Subscription Window	E1	NO/TM	0..1	The time interval during which the subscription is valid, where the subscription is associated with the successfully registered PurchaseItem/PurchaseData pair identified by the <serviceID> sibling element of the <RegistrationResponseServiceExtension> element. For time-based subscriptions, the network SHALL include this element. For pay-per-view, the network MAY include this element.The terminal MAY use this information to determine the validity period of a subscription. Contains the following attributes: startTime endTime	
startTime	A	NM/TM	1	NTP timestamp expressing the start of subscription.	unsignedInt
endTime	A	NO/TM	0..1	NTP timestamp expressing the end of subscription. This attribute SHALL NOT be present for open-ended subscriptions.	unsignedInt

Table 22: Structure of RegistrationResponseServiceExtension

5.1.6.8 LTKM Request Procedure

Upon the completion of the subscription/purchase transaction (as defined by the Service Request messages in Section 5.1.5.2), or once the lifetime of the current SEK/PEK in the Smartcard has expired, the required new SEK/PEK can be obtained via the LTKM Request procedure. This can occur:

- When the BCAST Terminal has missed a SEK/PEK key update procedure, due to, for example, being out of coverage;
- In response to a BM-SC solicited pull procedure.

For the Smartcard Profile, this procedure is the MBMS MSK request procedure as defined by [3GPP TS 33.246], in which the key identification information comprises a list of one or more Key Domain ID – SEK/PEK ID pairs, subject to the following clarification. For the (U)SIM Smartcard Profile terminal, the SRK used in the HTTP digest authentication of the subscriber corresponds to the MBMS Request Key (MRK); for the (R-)UIM/CSIM Smartcard Profile terminal, the SRK is the BCMCS Authentication Key (Auth-Key).

The terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

The BSM MAY include in the LTKM response one or several LTKM ResponseMSKExtensions in order to:

- include the LTKM(s) carrying the SEK(s)/PEK(s) requested in the LTKM request. This mechanism is defined in sections 5.1.6.8.1 and 5.1.6.10.3.

5.1.6.8.1 LTKM Response MSK Extension

The LTKM Response payload is an “mbmsMSKResponse” message defined according to XML schema “urn:3GPP:metadata:2005:MBMS:mskResponse” specified in section 11.8.1 of [3GPP TS 26.346 v7].

To allow the inclusion of BCAST-specific information at <Response> level of LTKM Response payload (i.e. at the level corresponding to one requested SEK/PEK), an *LTKMResponseMSKExtension* element is defined in the namespace “urn:oma:xml:bcast:pr:orderqueries:1.0” [BCAST10-XMLSchema-orderqueries]. This element MAY be included in each/any <Response> element in the response. When included in a <Response> element, it SHALL be present once as a child of <Response> element matching the <xs:any> wildcard defined there.

This *LTKMResponseMSKExtension* element is structured as follows:

Name	Type	Category	Cardinality	Description	Data Type
LTKMResponseMSKExtension	E		0..1	Defines a container for the inclusion of BCAST-specific information at the <mbmsMSKResponse> level of LTKM Response payload defined in section 11.8.1 of [3GPP TS 26.346 v7]. Contains the following attributes: version Contains the following sub-elements: LTKM	
version	A	NM/ TM	1	Version of this extension element. 0x00 identifies BCAST 1.0	unsignedByte
LTKM	E1	NO/ TO	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message) carrying the SEK/PEK identified by the <MSK> element sibling of <LTKMResponseMSKExtension> element. This element SHALL NOT be included if <ResponseCode> element sibling of <LTKMResponseMSKExtension> does not indicate status code “200 OK”. More details on this element are further specified in section 5.1.6.10.3.	base64Binary

Table 23: Structure of LTKMResponseMSKExtension

5.1.6.9 Deregistration Procedure

The Deregistration procedure is invoked by the terminal upon termination or suspension of the BCAST Client, or whenever the terminal wishes to indicate that it is not anymore available to receive LTKMs.

It may happen that the terminal is unable to perform a Deregistration procedure upon termination of the BCAST Client, due to uncontrolled power down or loss of coverage of interaction channel network. In this case, the terminal SHOULD perform a Deregistration procedure on the first occasion, i.e. upon next availability of interaction channel network while BCAST Client is besides not running (since otherwise the terminal would perform a Registration procedure).

For the Smartcard Profile, this procedure is the MBMS User Service Deregistration procedure as defined by [3GPP TS 33.246], in which one single MBMS User Service ID is indicated in the Deregistration Request: the value is "oma-bcast-allservices". This procedure is not applicable in the case of the (R-)UIM/CSIM Smartcard Profile, i.e., when BCMCS is the underlying BDS.The terminal SHALL send the Deregistration request to the PermissionsIssuerURI used by the terminal to send latest Registration request with this BSM, with "requesttype" parameter set to "deregister" instead of "register". This implies in particular that a BSM SHOULD accept De-Registration requests against a particular PermissionsIssuerURI as long as there are terminals considered in registered state against that URI. It is thus possible that the terminal might not receive any response to a De-Registration request after the BSM has invalidated the PermissionsIssuerURI.

The BSM SHALL interpret the Deregistration Request as a deregistration to the total list of (Purchase Items, PurchaseData) pairs for which the terminal is authorized to receive the related LTKMs.

The BSM SHALL include in the Deregistration Response:

- Either one "Response" element, in which the MBMS User Service ID value is "oma-bcast-noservices", in the case where the terminal is not subscribed to any purchase items
- Or one "Response" element, in which the MBMS User Service ID value is "oma-bcast-allservices".
- Or one or more"Response" elements in which each MBMS User Service ID SHALL be identified by the concatenation of GlobalPurchaseItemID and PurchaseDataReference values. These response elements SHALL contain the total list of MBMS User Service IDs for which the terminal is authorized to receive the related LTKMs.

The terminal SHALL handle the PDP context used by this procedure as specified in section 5.1.6.12.

In the latter case, the BSM MAY include in the deregistration response one or several DeregistrationResponseServiceExtensions, in order to:

- deliver LTKMs corresponding to the services that the terminal has deregistered to. This mechanism is defined in sections 5.1.6.9.1 and 5.1.6.10.3. The LTKMs contained in the deregistration response MAY be used to invalidate SEKs/PEKs, e.g. by carrying invalid Key Validity Data.

5.1.6.9.1 Deregistration Response Service Extension

The Deregistration Response payload follows the format of Registration Response payload: it is an "mbmsSecurityRegisterResponse" message defined according to XML schema "urn:3GPP: metadata:2005:MBMS:securityRegistrationResponse" specified in section 11.7.1 of [3GPP TS 26.346 v7].

To allow the inclusion of BCAST-specific information at the <Response> level of Deregistration Response payload (i.e. at the level corresponding to one deregistered service), a *DeregistrationResponseServiceExtension* element is defined in the namespace "urn:oma:xml:bcast:pr:orderqueries:1.0" [BCAST10-XMLSchema-orderqueries]. This element MAY be included in each/any <Response> element in the response. When included in a <Response> element, it SHALL be present once as a child of <Response> element matching the <xs:any> wildcard defined there.

This *DeregistrationResponseServiceExtension* element is structured as follows:

Name	Type	Category	Cardinality	Description	Data Type
DeregistrationResponseServiceExtension	E		0..1	Defines a container for the inclusion of BCAST-specific information at the <Response> level of Deregistration Response payload defined in section 11.7.1 of [3GPP TS 26.346 v7]. Contains the following attributes: version Contains the following sub-elements:	

				LTKM	
version	A	NM/ TM	1	Version of this extension element. 0x00 identifies BCAST 1.0	unsignedByte
LTKM	E1	NO/ TO	0..N	Smartcard profile BCAST LTKM (base64-encoded MIKEY message) associated to the service successfully deregistered to identified by <serviceID> element sibling of <DeregistrationResponseServiceExtension> element. This element SHALL NOT be included if <ResponseCode> element sibling of <DeregistrationResponseServiceExtension> does not indicate status code “200 OK”. More details on this element are further specified in section 5.1.6.10.3.	base64Binary

Table 24: Structure of DeregistrationResponseServiceExtension

5.1.6.10 LTKM delivery mechanisms

The BSM can send LTKMs over UDP to the terminal following BCAST-specific service provisioning messages (Service Response, Subscription Long-Term Key Renewal Response, Token Purchase Response, Unsubscribe Response) or MBMS-based provisioning messages (Registration response, Deregistration response, LTKM response). The BSM can also push to the terminal unsolicited LTKMs over UDP, to update SEKs/PEKs. Finally, the BSM can push messages over UDP (called triggers in this section) to trigger the terminal to initiate a LTKM request procedure or a registration procedure.

The terminal as well as the BSM MUST support LTKM delivery over UDP.

There are however situations where the terminal is temporarily or permanently not reachable by UDP:

- temporarily if for instance the terminal is configured to release its PDP context shortly after an HTTP-based procedure with the BSM, including the registration procedure.

  Note: this configuration must be avoided if the number of maintained PDP contexts is not an issue for the network.
- permanently if for instance the terminal is attached to a private IP network behind a NAT , or if the terminal sends the registration request via an HTTP Proxy which modifies sender’s IP address.

To cope with these situations, other LTKM delivery mechanisms than UDP MAY be used, such as the inclusion of LTKMs in the HTTP response to a service provisioning request, as well as trigger delivery over SMS bearer.

Note: this version of the specification defines no tools enabling terminal or BSM to detect network configurations problematic for the reliability of trigger/LTKM delivery over UDP (e.g. NAT equipments, HTTP Proxy modifying sender’s IP address, short-term PDP contexts, etc). For these problematic network configurations the following is advised:

- If UDP delivery issues are of permanent nature (e.g. NAT equipments), SMS bearer can be used for trigger delivery, and HTTP bearer for LTKM and Parental Control Message delivery.

If UDP delivery issues are only about short lifetime of PDP contexts, UDP bearer can still be used for trigger, LTKM and Parental Control Message delivery, provided that terminal and BSM are able to negotiate PDP context lifetime during Registration procedure.

5.1.6.10.1 Signaling of supported delivery mechanisms for triggers and LTKMs

The terminal SHALL indicate in the registration request the complete list of LTKM delivery mechanisms it will support starting from the time of this registration request till next registration request. This indication applies to all the LTKMs the BSM will deliver to the terminal during this period, whether these LTKMs actually carry a SEK/PEK or not (i.e. with KEMAC Encr Data Len = 0).

The terminal SHALL indicate in the registration request the complete list of trigger delivery mechanisms it will support starting from the time of this registration request till next registration request. Triggers in scope are messages initiating a BSM solicited pull procedure, and BSM solicited pull messages initiating a registration procedure.

The terminal SHALL indicate these supported delivery mechanisms by including in the registration request one <RegistrationRequestExtension> element containing one <LTKMDelivery> element, itself containing zero or more <Trigger> sub-elements to denote trigger delivery mechanisms, and one or more <Type> sub-elements to denote LTKM delivery mechanisms.

The terminal MAY however omit in the request the indication of supported delivery mechanisms, if it supports no more and no less than UDP and SMS bearers for trigger delivery, and UDP bearer for LTKM delivery.

The BSM SHALL handle this terminal indication as follows:

- For each successfully authenticated registration request it receives, the BSM SHALL determine which trigger and LTKM delivery mechanisms the terminal will support from this registration request till next registration request:
  - If <RegistrationRequestExtension> element is present and includes an <LTKMDelivery> sub-element, the BSM SHALL read terminal-supported trigger and LTKM delivery mechanisms from respectively <Trigger> and <Type> sub-elements.
  - Otherwise the BSM SHALL conclude that the terminal supports UDP and SMS bearers for trigger delivery, and UDP bearer for LTKM delivery.
- If the BSM supports one or more of the terminal-supported LTKM delivery mechanisms, the BSM SHALL include in the registration response a <RegistrationResponseExtension> element, and this element SHALL include an <LTKMDelivery> sub-element listing all the terminal-supported mechanisms which the BSM plans to use for further trigger and LTKM deliveries to this terminal, starting from this registration response.
  - The BSM MAY choose to not return an <LTKMDelivery> sub-element to implicitly signal to the terminal it only plans to use UDP bearer for trigger and LTKM delivery.
  - For this terminal, the BSM SHALL NOT later on use trigger and LTKM delivery mechanisms other than those implicitly or explicitly signaled to the terminal in the registration response.
  - If the BSM supports none of the terminal-supported LTKM delivery mechanisms (regardless of supported trigger delivery mechanisms), the BSM SHALL signal this to the terminal by a “403 Forbidden” in the HTTP status line of the response.
  - The BSM SHALL NOT attempt to deliver triggers or LTKMs to this terminal, from this registration response included, till next registration request.

<RegistrationRequestExtension> element and related <LTKMDelivery> sub-element are defined in section 5.1.6.7.1.

<RegistrationResponseExtension> element and related <LTKMDelivery> sub-element are defined in section 5.1.6.7.2.

5.1.6.10.2 LTKM delivery over SMS

In this version of specification, LTKM delivery over SMS designates the delivery of an LTKM initiating a BSM solicited pull procedure (specified in section 6.6.1 of [BCAST10-ServContProt]) or BSM initiated registration procedure (specified in section 6.6.2 of [BCAST10-ServContProt].

5.1.6.10.3 LTKM delivery over HTTP

The terminal MAY support LTKM delivery over HTTP as defined in this section.

In this version of specification, LTKM delivery over HTTP designates the delivery of LTKMs:

- in the Registration Response payload, by the inclusion of RegistrationResponseServiceExtension(s) and related <LTKM> sub-element(s), as defined in section 5.1.6.7.3.
- in the LTKM Response payload, by the inclusion of LTKMResponseMSKExtension(s) and related <LTKM> sub-element(s), as defined in section 5.1.6.8.1.
- in the Deregistration Response payload, by the inclusion of DeregistrationResponseServiceExtension(s) and related <LTKM> sub-element(s), as defined in section 5.1.6.9.1.
- in the Service Response payload, by the inclusion of <LTKM> elements in the Smartcard Profile specific part of the message, as defined in section 5.1.5.2.2.
- in the Unsubscribe Response payload, by the inclusion of <LTKM> elements in the Smartcard Profile specific part of the message, as defined in section 5.1.5.4.2.
- in the Token Purchase Response payload, by the inclusion of <LTKM> elements in the Smartcard Profile specific part of the message, as defined in section 5.1.5.5.2

The following applies for the delivery of LTKMs in any of these HTTP responses:

- The BSM SHALL NOT include LTKMs in unsuccessful HTTP responses.
- The BSM SHALL NOT include LTKMs initiating a BSM solicited pull procedure or BSM solicited pull messages initiating a registration procedure
- In case multiple LTKMs are carried in the same HTTP payload, the BSM SHALL insert them in order of increasing TS.

5.1.6.10.4 LTKM general processing

Unless otherwise stated, the terminal SHALL process all the LTKMs delivered by the BSM using any of the delivery mechanisms signaled by the BSM in the registration response, or using UDP if the BSM omitted this signaling in the registration response. The terminal MAY ignore LTKMs delivered by the BSM using other delivery mechanisms. Note that as the terminal signals the LTKM delivery mechanisms that it supports in the registration request, the BSM SHOULD NOT deliver LTKMs using a mechanism that is not supported by the terminal.

In case multiple LTKMs are carried in the same payload, the terminal SHALL process them one by one in order of inclusion in the payload.

For each processed LTKM respectively with V flag in HDR set or leading to the return of LTKM reporting message(s), the terminal SHALL send respectively one verification message or LTKM reporting message(s) over UDP to the BSM IP address resolved from NAF FQDN encoded in IDi payload, regardless of the method used by the BSM to deliver the LTKM.

Determination of the destination port to use is defined as follow:

- In case the LTKM is delivered over UDP, the terminal SHALL send the message to the same destination port as the destination port that was used by the BSM to deliver the LTKM.
- In case the LTKM is delivered via HTTP in the context of BCAST delivery mechanisms as specified in section 5.1.6.10.3the terminal SHALL send the message to destination port 4359

In case multiple LTKMs are carried in the same payload, the verification messages SHALL be sent one by one in order of LTKM processing.

5.1.6.11 Parental control messages delivery mechanisms

Potential problems for the delivery over UDP of parental control as described in Section 5.1.6.10 are also possible; therefore, the mechanisms for the delivery of the parental control messages are identical to those defined for the delivery of LTKMs in Section 5.1.6.10.

The terminal as well as the BSM MUST support parental control message delivery over UDP. To cope with situations where the terminal is not reachable, other delivery mechanisms than UDP MAY be used, such as the inclusion of parental control message in the HTTP response.

5.1.6.11.1 Signaling of supported Parental Control Message delivery mechanisms

Signaling of supported parental control messages delivery mechanisms SHALL be done as defined for the LTKM delivery mechanisms in section 5.1.6.10.1. The signaled mechanisms for LTKM delivery according to that section SHALL also be used for the delivery of Parental control messages.

5.1.6.11.2 Parental Control Message delivery over HTTP

The terminal MAY support delivery over HTTP as defined in this section.

In this version of specification, delivery over HTTP designates the delivery of parental control messages:

- in the Registration Response payload, by the inclusion of base64-encoded Parental control messages into the RegistrationResponseExtension as defined in section 5.1.6.7.2.

The following applies for the delivery of parental control messages in any of these HTTP responses:

- The BSM SHALL NOT include parental control messages in unsuccessful HTTP responses.

5.1.6.11.3 Parental Control Message general processing

The terminal SHALL process the parental control message delivered by the BSM following processing methods defined for LTKMs in section 5.1.6.10.4

5.1.6.12 PDP context handling

The terminal needs to create or reuse a Packet Data Protocol (PDP) context [3GPP TS 23.060] to perform unicast IP communications with the BSM.

For the (U)SIM Smarcard profile, the BSM can besides make use of this PDP context to deliver messages over UDP to the terminal, such as: LTKMs, BSM solicited pull procedure initiation messages, BSM solicited pull messages and Parental Control messages. The BSM can send these messages to the terminal over UDP subsequently to an HTTP procedure, or at the initiative of the BSM (unsolicited push). This implies that the BSM must know terminal IP address, as well as validity of this IP address in the time, which can be achieved using the following rules:

- Rules on PDP context below apply to the HTTP Digest authenticated procedures performed between terminal and BSM (Registration, Deregistration, LTKM Request, Service Request, Token Purchase, Unsubscription, Pricing Information and Account Inquiry).
- When performing one of these procedures with the BSM, the terminal SHALL reuse any existing active PDP context which it has previously used to communicate with this BSM. This rule is to ensure that the BSM sees at most one valid IP address for a given registered terminal (B-TID).
- The BSM SHALL infer terminal IP address from the latest of these procedures successfully performed by the terminal with this BSM.

  Note: although Pricing Information and Account Inquiry procedures are not subject to subsequent LTKM delivery, this rule applies also to these procedures, as updating terminal IP address whenever possible reduces the likelihood for the BSM to send messages over UDP to an IP address reallocated to another terminal (case of terminal not being able to deregister before the release of its PDP context).

- The minimal lifetime of PDP context used by these procedures SHALL be negotiated at the time of each terminal registration and re-registration with the BSM (either explicitly via the inclusion of <PDPContextLifetime> element, or implicitly via the omission of this element, in Registration Request and/or Registration Response) and SHALL apply from this (Re-)registration procedure (included) till the Deregistration procedure with this BSM (included).
- For Registration, LTKM Request, Service Request, Token Purchase and Unsubscription procedures:
    - If <PDPContextLifetime> element was absent in Registration Response, the terminal SHALL NOT release the PDP context used by this procedure until completion of next Deregistration procedure or until a PDP context deactivation procedure has been initiated by the network as defined in [3GPP TS 23.060].
    - If <PDPContextLifetime> element was present in Registration Response and set to zero, the terminal MAY release the PDP context used by this procedure immediately after completion of the HTTP message flow and the delivery of any requested Parental Control verification, LTKM verification and LTKM reporting messages.
    - If <PDPContextLifetime> element was present in Registration Response and set to a number of seconds greater than zero, the terminal SHALL NOT release the PDP context used by this procedure until this number of seconds has elapsed (starting from completion of the HTTP message flow) or until completion of next Deregistration procedure, or until a PDP context deactivation procedure has been initiated by the network as defined in [3GPP TS 23.060].
- For the Deregistration procedure:
    - The terminal MAY release the PDP context used by this procedure immediately after completion of the HTTP message flow and the delivery of any requested LTKM verification and LTKM reporting messages.
- For Pricing Information or Account Inquiry procedure:
    - If the PDP context used by this procedure was constrained to a specific lifetime by a previous HTTP procedure, the terminal SHALL NOT release the PDP context used by this procedure until this lifetime has expired or a PDP context deactivation procedure has been initiated by the network as defined in [3GPP TS 23.060].
    - Otherwise the terminal MAY release the PDP context used by this procedure immediately after completion of the HTTP message flow.

5.1.7 Message Compression

The Service Provisioning messages MAY be compressed during the transport of the messages. In that case, the compression applies to entire Service Provisioning message which is the payload of HTTP message. If the compression is used, in the HTTP message delivering the Service Provisioning message the “Content-Encoding” attribute SHALL be present in the HTTP header and set to MIME value representing the compression scheme.

The BSP-M in the BSM SHALL support the GZIP algorithm for the delivery of Service Provisioning messages. The BSP-C in the Terminal SHALL support the GZIP algorithm for the delivery of Service Provisioning messages. In case GZIP compression is used for the delivery of Service Provisioning messages, the “Content-Encoding” attribute SHALL be set to “gzip”.

5.1.8 Provisioning Trigger Message (DRM Profile only)

The message below is defined to trigger the terminal to send a provisioning message to the BSM. A Terminal and a BSM supporting the DRM profile and the Web-based Service Provisioning feature SHALL support the Provisioning Trigger Message.

The BSM SHALL use SMS to send this message to the Terminal. The SMS SHALL satisfy the following conditions:

- The SMS carries a WAP connectionless push (WDP/WSP encoding) as defined in [OMA Push].
- The WSP content type header contains the Content Type code registered by OMNA for ‘application/vnd.oma.bcast.provisioningtrigger’ (see Appendix H.6,) i.e. the binary value 0x031B.

- The WSP X-Wap-Application-Id header contains the binary code registered by OMNA for the PUSH Application ID identifying the BCAST Push client, as specified in section 9 of [BCAST10-Distribution].

The message SHALL be structured as follows. Note that the 'type' parameter signals the type of the message and as such determines its structure (i.e. number, semantics and size of the parameters contained).

Data Field Name	Data Type
Provisioning_Trigger_Message {	
type	uimsbf8
if(type==0) {	
LTKMRenewalRequestTrigger.idCode	uimsbf8
LTKMRenewalRequestTrigger.url	bytestring
}	
}	

Table 25: DRM Profile Trigger Message Structure

uimsbfN	Unsigned Nbit Integer, most significant bit first
bytestring	Array of bytes

Table 26: Mnemonics used in Table 25

type	Signals the type of the message. 0 – BCAST 1.0 LTKMRenewalRequest Trigger Message 1-255 – reserved for future use Terminals MAY discard messages with an unknown value in the 'type' field.
LTKMRenewalRequestTrigger.idCode	Code signalling the string to put into the 'globalIDRef' attribute 0 – "oma-bcast-allservices" 1 – "oma-bcast-newservices" 2-255 – reserved for future use
LTKMRenewalRequestTrigger.url	Signals the URL to which to send the LTKMRenewalRequest message as a null-terminated string

Table 27: Semantics for Table 25

If the terminal receives a message with the 'type' parameter equals to 0, it SHALL send an LTKMRenewalRequest to the BSM addressed by the URL signalled in the parameter 'LTKMRenewalRequestTrigger.url', containing one instance of 'PurchaseItem' with the 'globalIDRef' attribute set to the value signalled by the parameter 'LTKMRenewalRequestTrigger.idCode'. This message SHALL NOT be sent if the URL is not available to the terminal as 'purchaseURL' in any 'PurchaseChannel' fragment in the Service Guide, and is also not trusted by the terminal based on some other mechanism outside the scope of this specification.

5.1.9 Web-based Service Provisioning

A Terminal and server MAY support Service Provisioning via a web-portal. The description of web portal provisioning is based on the following assumptions:

- The web portal is a completely separate entity from the BSM (NAF), BSF, etc. and has no knowledge of key management.
- No HTTP digest authentication as per [3GPP TS 33.246] or [3GPP2 X.S0022-A](used in the Smartcard profile service provisioning messages) is required by the portal.

The Terminal MAY receive additional information related to the PurchaseItem, PurchaseData, and PurchaseChannel fragments using the '*url*' attribute of the '*extension*' element in each fragment. The Terminal SHALL use the '*PortalURL'* element of the PurchaseChannel fragment, defined in the Service Guide, as the entry point for Service Provisioning via a web portal. The *PortalURL* can be used to support three purposes:

1. The *PortalURL* provides additional information on services available over this PurchaseChannel. This method SHALL be signalled by setting the attribute *supportedService* under *PortalURL* to "0". In this case the Terminal MAY access the *PortalURL* to retrieve information on supported services but SHALL NOT purchase or (un)subscribe to the services by accessing the URL. In this case, the service provisioning functions SHALL be achieved by addressing Service Provisioning messages to the *PurchaseURL* as defined in section 5.1.5.

2. The *PortalURL* supports the full set of service provisioning functionality via the web-portal in addition to providing service related information. This method SHALL be signalled by setting the attribute *supportedService* under *PortalURL* to "1". The Terminal SHALL access the *PortalURL* where the Terminal SHALL expect that the facilities for service provisioning are provided by the web-portal. When the *supportedService* attribute under *PortalURL* is set to "1", the Service Provisioning messages sent to the *PurchaseURL* as defined in section 5.1.5 SHALL NOT be used.

3. The *PortalURL* provides additional information on services available over this PurchaseChannel. The Terminal MAY achieve the service provisioning either via web-portal or by addressing Service Provisioning messages to the *PurchaseURL* as defined in section 5.1.5. This method SHALL be signalled by setting the attribute *supportedService* under *PortalURL* to "2".

Note: care must be taken when value "2" of 'supportedService' is used that the web-portal and the BSM are correctly synchronized, as synchronization delays can result in the user being subscribed twice to the same service.

In the context of the above three methods, there are three ways the request to *PortalURL* can be formed.

1. Request without reference to a specific PurchaseItem.

When Terminal accesses the *PortalURL* without any specific reference to any PurchaseItem, the Terminal SHALL issue an HTTP POST request to the *PortalURL*. This request SHALL be made over SSL/TLS when "https:" scheme is present in PortalURL. This request SHALL besides follow 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. For example, if *PortalURL* is http://server.example.org/webshop". The HTTP POST request sent to the web portal would be "http://server.example.org/webshop", not containing any associated data block.

2. Request with reference to a specific PurchaseItem.

When the Terminal accesses the *PortalURL* with specific reference to a PurchaseItem or a set of PurchaseItems, the Terminal knows the relevant globalPurchaseItem IDs from the Service Guide.

3. Request with reference to a specific PurchaseItem and associated PurchaseData. In similar fashion than method 2, the terminal knows the identifier of the relevant PurchaseData fragments from the Service Guide.

For methods 2 and 3 defined above, the Terminal SHALL issue an HTTP POST request to the *PortalURL*. This request SHALL be made over SSL/TLS when "https:" scheme is present in PortalURL.This request SHALL besides follow 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. The PurchaseItem(s) are identified using the globalPurchaseItem_ID(s). Each globalPurchaseItem_ID SHALL be signalled in a separate name-value pair, using "globalPurchaseItemID" as the name. The PurchaseData fragments are identified using their 'id' attribute, each PurchaseData fragment id SHALL be signalled in a separate name-value pair, using "purchaseDataID" as the name. For example, if *PortalURL* is "http://server.example.org/webshop" and the globalPurchaseItemIDs are "aau17135@bsda.example.org" and "fhh7982@bsda.example.org" and "jke132486@bsda.example.org", and there is also a related PurchaseData fragment id "bbu17135@bsda.example.org", the HTTP POST request sent to the web portral would be "http://server.example.org/webshop", containing a data block of the following structure:

"globalPurchaseItemID=aau17135@bsda.example.org&
globalPurchaseItemID=fhh7982@bsda.example.org&
globalPurchaseItemID =jke132486@bsda.example.org&
purchaseDataID= bbu17135@bsda.example.org"

NOTE 1: it is reminded that, according to [BCAST10-SG], the PurchaseData fragment points to one, and only one, PurchaseItem fragment. This allows mapping the purchaseDataID with the correct globalPurchaseItemID upon processing the request.

NOTE 2: “globalPurchaseItemID” name is intentionally reused for each name-value pair. This reuse is conformant with [HTML4.01] and the web-based system is assumed to support it.

The Terminal MAY receive an HTTP response message that contains a list of PurchaseItems, each of which is associated with either price information or price information and purchase options. Price information for each listed PurchaseItem SHOULD be consistent with that in the relevant PurchaseData fragment announced in the Service Guide. However, user specific purchase options such as promotions could be included in the response. The implementation and display of user specific purchase options is out of scope for BCAST 1.0.

After a successful subscription or purchase event, the BSM SHALL send a Trigger message to the terminal. The Trigger message and its further processing differs in the DRM and Smartcard profiles

For the DRM Profile:

1. Once the web-based subscription/purchase transaction is completed, the web-based system informs the BSM of the completed transaction via means that are outside the scope of this specification.
2. The BSM SHALL send a Provisioning Trigger message (see section 5.1.8) the Terminal, providing a URL to which the terminal can send the subsequent provisioning message, setting ‘type’=0 and ‘LTKMRenewalRequestTrigger.idCode’=1, i.e. ‘oma-bcast-newservices’.
3. . The Terminal SHALL process this message as specified in section 5.1.8) and send an LTKMRenewalRequest to the BSM.
4. The BSM SHALL respond with an LTKMRenewalResponse that contains all those PurchaseItem/PurchaseData combinations for which the Terminal has not yet received a ROAP trigger, plus a ROAP trigger that allows the terminal to acquire those keys.

The BSM MAY re-send the trigger message if the terminal does not react to it within an assumed time interval. The terminal MAY send the LTKMRenewalRequest with ‘globalIDRef’ set to “oma-bcast-newservices” without having received a trigger message. The Terminal MAY ignore a trigger message if another trigger message with identical parameters has been previously received within a short time frame and successfully processed.

For the Smartcard Profile:

1. Once the web-based subscription/purchase transaction is completed, the web-based system informs the BSM of the completed transaction via means that are outside the scope of this specification.
2. If the terminal is in a registered state in BSM, and if at least one LTKM trigger bearer (e.g. UDP or SMS) has been successfully negotiated between BSM and terminal, the BSM SHALL send a BSM solicited pull message to the terminal to trigger registration procedure and subsequent delivery of LTKMs, including at a minimum the LTKMs associated to the items just purchased on the web-based system.

3. Otherwise, the BSM SHALL wait for another opportunity to deliver these LTKMs, such as the registration procedure initiated by terminal on BCAST application start.

In case of the other subsequent operations such as LTKM renewal, Token Purchase, Account Inquiry, the Terminal SHOULD use either the general service provisioning procedures or Smartcard Profile Service Provisioning procedures, defined in Sections 5.1.5 and 5.1.6 respectively, according to the security profile. The Terminal MAY unsubscribe using the web portal or using the General Service Provisioning procedure defined in section 5.1.5.

5.2 Terminal Provisioning

The Terminal Provisioning function SHALL support OMA Device Management [OMA DM], as specified in this chapter. To allow firmware upgrades using DM over the interaction channel, the Terminal Provisioning function SHOULD support OMA FUMO 1.0 [OMA FUMO].

Terminal Provisioning function provides data structures to provision and manage the terminal through the interaction channel using OMA DM [OMA DM].

The interfaces related to Terminal Provisioning function, as outlined in BCAST Architecture [BCAST10-Architecture] are normatively specified as follows:

- Over interface TP-7, both the network and the terminal SHALL support exchange of terminal provisioning and management messages as specified in [OMA DM]

5.2.1 Terminal Provisioning of BCAST Client

The Terminal Provisioning Client Component (TP-C) SHALL receive the parameters needed for OMA BCAST service (see [BCAST10-Services] Appendix F) by the Terminal Provisioning function which manage the terminal configuration parameters, e.g. data, parameters and applications with the help of OMA DM [OMA DM]. This information would be delivered through TP-7 as specified in OMA DM [OMA DM].

The Terminal Provisioning Client Component (TP-C) SHALL be able to:

- receive the parameters needed for BCAST service included in the terminal provisioning messages sent over TP-7.
- update the parameters needed for BCAST service included in the terminal provisioning messages sent over TP-7.
- perform firmware upgrades of the BCAST client using the interaction channel over TP-7.

Further, the existence and access description to Terminal Provisioning function MAY be declared through the Service Guide using the Service, Access and Content fragments of Service Guide or through the process as specified in OMA DM. Both of the following cases are specified in section 5.2.2:

- Declaration of the existence and access to the OMA DM based exchange over TP-7.

5.2.2 Declaring the existence of and access to Terminal Provisioning

There are two ways to declare the existence of and the access to Terminal Provisioning with Service Guide: Terminal Provisioning declared as a Service; and; Terminal Provisioning declared as a means for accessing of a Service. The terminal SHALL support both methods of declaring the Terminal Provisioning within the Service Guide. The following sections specify both of these ways.

The TP-C MAY also be bootstrapped with the Terminal Provisioning server information to access the Terminal Provisioning by TP-7.

5.2.2.1 Declaring Terminal Provisioning as a Service within Service Guide

When the Terminal Provisioning is declared as a service, the following applies:

- There SHALL be at least one Service fragment with the value of attribute “ServiceType” equals “9 - Terminal Provisioning service”.
- There SHALL be at least one Access fragment that specifies the access to the above-mentioned Service:
  - In case Terminal Provisioning over TP-7 is declared, the AccessType SHALL contain “UnicastServiceDelivery” element, which defines the access to the respective provisoning server.
- There MAY be one or more Content fragments that specify the Terminal Provisioning messages as files, as defined in section 5.2.1.

5.2.2.2 Declaring Terminal Provisioning as an Access of a Service within Service Guide

When the Terminal Provisioning is declared as an access of a service, the following applies:

- There SHALL be at least one Service fragment that defines a service of arbitrary type.
- There SHALL be at least one Access fragment associated with the above-mentioned Service. The Access fragment SHALL have “ServiceClass” element present with value “urn:oma:bcast:oma_bsc:tp:1.0”. Further:
  - In case Terminal Provisioning over TP-7 is declared, the AccessType SHALL contain “UnicastServiceDelivery” element, which defines the access to the respective OMA DM server.

5.2.2.3 Declaring Terminal Provisioning through Bootstrap

5.2.2.3.1 Initiation of Terminal Provisioning by DM server

Terminal Provisioning through bootstrap (e.g. server information or account for such as the Session Description, Authentication, and/or Connectivity) MAY be supported as specified in [OMA DM]. Bootstrap information comprising DM server’s Connectivity information, would be delivered to the terminal. Then, the DM server would deliver to the terminal information for the Terminal Provisioning server such as Session Description, Authentication Information (certificate, OCSP Response) for secure delivery and/or Connectivity as specified in [OMA DM].

5.2.2.3.2 Initiation of Terminal Provisioning by Smartcard

Terminals with cellular interface and (U)SIM/R-UIM/CSIM that support BCAST and OMA DM [OMA DM] SHALL support bootstrap from the smartcard as specified in [DMBOOT]. In these terminals DM TND Serialization [DMTNDS] SHALL also be supported otherwise

The following table shows the DM Client Requirements. The table is based on section 8 of [ERELDSC].

Feature / Application	Status	References
DM Client	MANDATORY	[DMPRO] [DMREPU] [DMSEC] [DMTND] [DMSTDOBJ] [DMDDFDTD]
DM Client Bootstrap	MANDATORY if Terminal with cellular radio interface and (U)SIM/(R-)UIM/CSIM	[DMBOOT]
DM TND Serialization	MANDATORY if Terminal with cellular radio interface and (U)SIM/(R-)UIM/CSIM	[DMTNDS]

Table 28: OMA BCAST Device Management Client Requirements

5.2.3 Carrying OMA DM messages through Interaction Channel

Over interface TP-7, DM provisioning messages SHALL be delivered using DM mechanism. The details follow the OMA DM procedure.

5.3 Interaction

The BCAST enabler specifies different types of interactions between the end user and their terminal, and the service provider.

These are the following:

1. Interactive retrieval of the Service Guide (SG). The terminal requests, and receives, the service guide or changed parts of the service guide for a service. This type of interaction is described in the [BCAST10-SG], section 5.4.3.
2. Interactive retrieval of additional information related to Service Guide fragments, for example in form of a webpage presenting additional information. This is enabled using the ExtensionURL which can optionally be included into some SG fragments for retrieving further information about the fragment by accessing the URL. For details see in the [BCAST10-SG].
3. Service interaction, i.e. interaction as part of the service (in contrast to the previous two types of interaction, which are used to receive information about a service). Examples for such interactions within a service are voting about the service or actor, or the offer to the user to order a ring-tone matching the music that is just played in a show. This is enabled using interactivity information in the SG as an entry point and interactivity media that are distributed in a channel associated with the service itself. This is described in more detail in this document in section 5.3.6.
4. Interactive delivery of BCAST services, i.e. delivery over the interaction channel. This is enabled using the UnicastServiceDelivery in the SG.

In general, the availability of the interaction channel is assumed. However, the interaction channel may be temporarily unavailable, for example due to lack of radio coverage. Further, devices without access to an interaction channel are possible; however, such devices may have limited functionality.

5.3.1 Protocols and media codecs for Service Interaction Function

This section describes the protocols which are provided by the Service Interaction Function of the BCAST enabler at the interface between BSI-G and BSI-C through SI-8 and the media codecs the BCAST application supports.

With respect to the protocols, please note that this section only specifies the protocols to be used for the Service Interaction Function. The use of the interaction channel by other functions (e.g. Service Guide Function) is defined in the respective other parts of the BCAST specifications and is not part of this section.

The available interaction protocols for a service are signalled in the Service Guide according to section 5.1.2.4 in the BCAST Service Guide specification. If a terminal does not support any of the interaction protocols specified here, it SHALL not offer the interactive parts of the service to the user.

A service making use of the interaction function MAY use any of the following protocols.

Regarding support of the protocols in the terminal for use by the Service Interaction Function, the following rules apply:

- The terminal SHALL support the following protocols: IP, TCP, HTTP.
- The terminal MAY support the following protocols: SMS, IPSEC, UDP, MMS, WAP, HTTPS based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246], SIP/IMS.
- If a terminal supports SMS, it SHALL support SMS as an interaction protocol for BCAST services.
- If a terminal supports MMS, it SHALL support MMS as an interaction protocol for BCAST services.
- Furthermore, the terminal MAY offer the user an option to initiate a voice call from the service application of an interactive broadcast service. In this case, the terminal SHALL prompt the user before actually making the call.

5.3.2 Interactive retrieval of Service Guide

If the Terminal has a capability for interaction, it SHALL support interactive retrieval of Service Guide fragments as specified in [BCAST10-SG].

5.3.3 Interactive retrieval of Service related information

Within the Service Guide itself, the network MAY include an "ExtensionURL" element with any fragment. The semantics of this element is to provide a pointer to a web resource providing further information related to the fragment (For example, a www page related to the certain content can be reached by following an extension URL in Content fragment). If the Terminal has a capability for interaction, it SHALL support this element and SHALL be capable of accessing such additional information by using HTTP.

5.3.4 Interactive service ordering

After receiving Service Guide, Terminal can subscribe or purchase PurchaseItem via Interaction Network. Interactive service ordering includes service request for subscription or purchase, Subscription LTK Renewal request, Token purchase request and also unsubscription request specified in the section 5.1.6 of this specification.

5.3.5 Interaction for service and content protection

Service and Content Protection have four layers. Those four layers are the registration layer, the LTKM delivery layer, the STKM delivery layer and the traffic encryption layer. Terminal executes registration procedure with BSM to acquire Registration Data. After that Terminal acquires SEK and/or PEK from LTKM delivered from BSM or BSD/A. Terminal can perform traffic decryption using TEK after receiving STKM from BSD/A.

5.3.6 Service related interaction and feedback

The mechanism described in this section allows the user to interact with the service, for example for voting applications. The main entry point for interactivity services is the InteractivityData fragment in the SG (see section 5.1.2.10). This InteractivityData fragment points to one or more interactivity media documents, which contain the actual interactivity media objects.

5.3.6.1 Interactivity Media Document

An instance of 'InteractivityMediaDocument' represents details of an interactive component of a service. This component is thought as interactive means for a user to consume the service. The interactive component can consist of multiple instances of 'InteractivityMediaDocument' and can be associated to both services and individual pieces of services through the 'InteractivityData' fragment of the Service Guide. In practice, the contents of an 'InteractivityMediaDocument' triggers the Terminals to render the details of the interaction(s) "interactivity media objects" message onto the GUI which in turn is thought to prompt the user of the terminal to react on it..

5.3.6.1.1 Media Object Group and Media Object Set

Each instance of 'InteractivityMediaDocument' can consist of multiple media object groups, and each media group can consist of one or more media object sets. A media object set is a bundle of related media objects to be rendered as a unit (e.g. XHTML pages + external stylesheet + pictures) and clearly identified as pertaining to a specific interactivity technology (SMS, MMS template, XHTML…). From each media object group only one media object set is rendered at the same time by the terminal. This is indicated by the media object set with the highest relative priority, expressed by the element 'RelativePreference', and that is besides supported by the terminal. If a media object set is not supported by the terminal it is discarded. If none of the media object sets are supported by the terminal the terminal SHALL display the alternative text.

The media objects of a media object set are packed into one file bundle transported separately from the instances of 'InteractivityMediaDocument' (except for email and SMS). The element 'MediaObjectGroup' of InteractivityMediaDocument only describes each media set the involved interactivity technology, the type of included media objects, and the file delivery information needed to retrieve the set of media objects. This decoupled structure allows the terminal to discard the unsupported media object sets at the very beginning of file bundle reception, and more importantly to avoid storing them. Content promotion can be enabled by one media object group in the InteractivityMediaDocument . By referring to this same media object group through the attributeOnActionPointer of the element 'ActionDescriptor' the terminal will always return to the same media object set when the end-user triggered the terminal to send out a message over the interaction channel. Referring to information on an external web-site can be enabled by declaring one media object group with an XHTML MP media object set in the InteractivityMediaDocument . By omitting the attribute OnActionPointer of the element 'ActionDescriptor', the XHTML hyperlinks can refer the user to external web-sites. Further, SMS-URIs according to

[URI-Schemes] can also be embedded in XHTML. If the Terminal supports XHTML, it SHALL support SMS URIs [URI-Schemes].

Time-dependent behaviour of the interaction can be enabled by defining 3 media object groups in the InteractivityMediaDocument . The first media group defines the media to start with, e.g. a list of possible answers of a voting. When the user answers in time (as defined by the attribute InputAllowedTime of the element 'ActionDescriptor'), the user is presented the media object set from the second media group (as defined by the OnActionPointer). If the user waits too long or does not provide any input the media object set from the third media group is presented (as defined by the attribute OnTimeOutPointer of the element 'ActionDescriptor'). Setting the Update Flag in turn in an instance of 'InteractivityMediaDocument' having group position zero to "true" enables the rendering of the media object set for the next question. When the amount of time represented by the attribute InputAllowedtime is passed the terminal will start listening to the file delivery channel for an instance of InteractivityMediaDocument with a higher value of GroupPosition.

Interactivity Media Document 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. The explicit signaling of the required parameters in Interactivity Media Document prevails, for that interaction, over default values possibly signaled in the corresponding 'Interactivity Data' fragment.

Instances of 'InteractivityMediaDocument' and the files declared in the element 'MediaObjectSet' are delivered using BCAST File Distribution Function. The system MAY deliver instances of 'InteractivityMediaDocuments' and associated files over broadcast file distribution or serve those over interactive channel. Terminals supporting the interactive channel SHALL support reception of the instances of 'InteractivityMediaDocuments' and the associated files over broadcast file distribution.

5.3.6.1.2 Format of Interactivity Media Document

The following table defines the message format of Interactivity Media Documents. . The XML schema for this message format is defined in [BCAST10-XMLSchema-InteractivityMedia].

Name	Type	Category	Cardinality	Description	Data Type
InteractivityMediaDocument	E	NO/TM		The InteractivityMediaDocument defines the actual InteractivityMedia objects Contains the following attributes: groupID groupPosition id version validFrom validTo Contains the following sub-elements: MediaObjectGroup PrivateExt	
groupID	A	NM/TM	1	ID of the group of Interactivity Media document, globally unique	anyURI
groupPosition	A	NM/TM	1	Relative position of this document in the group. The greater value has higher priority to handle (i.e 2 has higher priority than 1).	unsignedShort
id	A	NM/TM	1	ID of the InteractivityMediaDocument , globally unique.	anyURI
version	A	NM/TM	1	Version of this InteractivityMediaDocument. The newer version overrides the older one with the same id as soon as it has been received.	unsignedInt
validFrom	A	NM/TM	0..1	The first moment when the media object sets in this document is valid to be rendered. If not given, the media object sets SHALL be rendered	unsignedInt

				as soon as they are available. This field expressed as the first 32bits integer part of NTP time stamps.	
validTo	A	NM/TM	0..1	The last moment when the media object set is valid to be rendered. If not given the rendering is assumed to end in undefined time in the future. This field expressed as the first 32bits integer part of NTP time stamps. Whenever there is an InteractivityMediaDocument available with the same GroupID but with a higher GroupPosition and the 'validFrom' time of that InteractivityMediaDocument arrives, the terminal SHALL stop rendering the current document and render the new InteractivityMediaDocument.	unsignedInt
MediaObject Group	E1	NM/TM	1..N	Grouping of the media object sets, which serve the same purpose during interactivity, e.g. as a starting media object set, as a media object set to be shown after action was taken or to be shown after time-out was reached. Has the following attributes: id startMediaFlag Has the following sub-elements: ActionDescriptor BackOffTiming MediaObjectSet SMSTemplate EmailTemplate VoiceCall Weblink Alternative text	
id	A	NM/TM	1	The ID of the media group	anyURI
startMediaFlag	A	NM/TM	1	The flag indicates, whether the media object sets in this MediaObject-Group should be started with. There SHALL only be one MediaObjectGroup with this flag set to "true" in an InteractivityMediaDocument	boolean
Action Descriptor	E2	NM/TM	0..1	The action descriptor describes the behaviour of the terminal when the media objects enable end-user input. Has the following attributes inputAllowedTime onTimeOutPointer updateFlag onActionPointer	
inputAllowed Time	A	NM/TM	0..1	The last moment the terminal allows the end-user to provide end-user input for the active media object set in this media object group. This field is expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
onTimeOutPointer	A	NM/TM	0..1	This pointer refers to the ID of a media object group in this InteractivityMediaDocument. When	anyURI

				the InputAllowedTime is passed the terminal SHALL present the appropriate media object set of the MediaObjectGroup indicated by the OnActionPointer. The terminal SHALL NOT present this media object set if the terminal has already presented the media object set indicated by the OnActionPointer.	
updateFlag	A	NM/TM	0..1	Whenever this flag is “true” the terminal shall listen to and fetch the following interactivity media document and the associated media objects from the file delivery stream when the Inputallowedtime is passed. The following interactivity media document is identified by the document with the same group ID and a higher GroupPosition number.	boolean
onActionPointer	A	NM/TM	0..1	This pointer refers to the ID of a media object group in this interactivity media document. When the end-user undertakes action before the InputAllowedTime, which triggers the terminal to send out a message over the interaction channel (e.g. MMS, SMS or HTTP request), the terminal SHALL present the appropriate media object set of the media object group indicated by this pointer. If this pointer refers to the same ID as the current media object group, the terminal SHALL return to the same media object set after completing the action. In this case InputallowedTime and OnTimeOutPointer SHALL NOT be declared.	anyURI
BackOffTiming	E2	NM/TM	0..1	This element specifies 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. If present, the interaction, if any, SHALL be sent back in the time interval [OffsetTime, OffsetTime+RandomTime] after the event that triggered the interactivity (e.g. user feedback). The exact time within the allowed time window shall be random with uniform probability. Explicit timing behaviour expressed in Interactivity Media Document prevails over possible default timing behaviour expressed in InteractivityData.	
offsetTime	A	NM/TM	1	The OffsetTime 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	The RandomTime refers to the time window length over which a device SHALL calculate a	decimal

				random time for the transmission of interaction. The method provides for statistically uniform distribution over a relevant period of time. 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.	
MediaObject Set	E2	NM/TM	0..N	A media object set is defines the media objects attached to one interactivity technology proposed in the MediaObjectGroup. These media objects are related to each other, and form an interactivity unit to be rendered upon MediaObjectGroup activation (provided this interactivity technology is the one selected for rendering). The set of media objects is not stored in the MediaObjectGroup itself (i.e. in the InteractivityMediaDocument) but as another external file, where this external file is : either one uncompressed media file (like a .3GP video, a .JPEG picture). or one GZIP archive file containing one or several compressed media objects (a .GZ file e.g. containing a compressed SMIL + 3GP video + text) The GZIP archive format is the one defined in [RFC 1951] and [RFC 1952]. In case the archive contains multiple media objects, it consists of the plain concatenation of each compressed media object (i.e. each GZIP member), as specified in section 2.2 of [RFC 1952]. The optional FNAME field SHOULD be set by the sender in each GZIP member header, with an FNAME value in accordance with the 'Object' Content-Location one (see below Content-Location description). The 'MediaObjectSet' element contains the following attributes: relativePreference …………Content-Type Content-Location The language of a MediaObjectSet element is expressed by using the built-in XML attribute xml:lang with this element. In case this attribute is not instantiated, the terminal SHALL interpret the MediaObjectSet element to be applicable for any language. The 'MediaObjectSet' element contains the following elements: Description Object	

				File	
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this media object set. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media object sets elements are instantiated in this 'MediaObjectGroup' then all the media object sets SHALL have mutually exclusive values of 'relativePreference'. If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all of these elements SHALL have the 'relativePreference' attribute instantiated. If only a single media object set is instantiated in 'MediaObjectGroup' then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt (32 bits)
Content-Type	A	NM/TM	1	Gives the media type of the 'MediaObjectSet''s external file : If this media type is 'application/x-gzip', the external file is a GZIP archive file containing one or several media objects. Otherwise (in this version of the specification) the external file is one uncompressed media file (e.g. 'video/3gpp' for a 3GP video file containing a SMIL presentation). In case the external file is transported by FLUTE, this attribute MUST match the 'File' Content-Type value provided in the FDT instance(s) describing this file.	string
Content-Location	A	NM/TM	1	Uniquely identifies the 'MediaObjectSet''s external file within the file delivery session. In case this external file is transported by FLUTE, this attribute MUST match the 'File' Content-Location value provided by the FDT instance(s) describing this file. Using this attribute, multiple 'MediaObjectSet' instances belonging to the same or different 'MediaObjectGroup' instances of the same or different instance of 'InteractivityMediaDocument' MAY point to the same external file.	anyURI
Description	E3	NM/TM	0..1	Description of the Media Object Set, expressed in the same language as the parent 'MediaObjectSet' element. This is used to provide the end-user extra information regarding the Media Object Set content.	string
Object	E3	NM/TM	0..N	Describe each media object contained in the media object set. Depending on 'MediaObjectSet''s external file nature: if a single uncompressed file, this element is not	

				needed unless it can provide supplemental information not given by parent ‘MediaObjectSet’ (such as ‘PartType’, etc.). if a GZIP archive, the sequence order of ‘Object’s in ‘MediaObjectSet’ MUST be the same as the sequence of members in the GZIP archive (side-by-side relationship between ‘Object’ sequence and GZIP members). Contains the following attributes: ……….Content-Location Content-Type ………..start Contains the following elements: PartType	
Content-Location	A	NM/TM	0..1	*If ‘MediaObjectSet’’s external file is an uncompressed file:* useless. *If ‘MediaObjectSet’’s external file is a GZIP archive:* The external file can be found by decompressing the n-th member of the GZIP archive, given n is the position of the ‘Object’ in the ‘MediaObjectSet’. The Content-Location value SHALL be a Relative-Path Reference as defined in [RFC 3986] and SHALL represent the sub-folder(s) + the filename of the deflated GZIP member to be used on storage. This relative storage content location is intended to be directly pointed by common markup language references (typically via src=”” and href””). If present, the FNAME field of the GZIP member MAY be verified against the filename part of Content-Location, ignoring case differences. In case these two values differ, the terminal MAY choose to discard the Media Object Set. When storing the deflated media object, the terminal MUST create any indicated sub-folder(s) specified in the Content-Location, and store the media object in the leaf sub-folder, using the file name indicated in the Content-Location.The terminal SHOULD preserve the letter case specified in the Content-Location value when deflating the subfolders and the media file locally. The dot-segment “.” MUST be supported. Content-Location value SHALL be unique within the sequence of ‘Object’ elements belonging to the same ‘MediaObjectSet’ in the following respect: A folder (including root folder) SHALL NOT contain two different subfolders or files for which the names only differ by the letter case. For security reasons, the terminal SHOULD	anyURI

				discard the Media Object Set in case a naming conflict is detected. For security reasons, the terminal SHOULD discard the Media Object Set if one or several dot-segments “..” are present in the Content-Location.	
Content-Type	A	NM/TM	1	*If ‘MediaObjectSet’’s external file is an uncompressed file:* useless (information already given in ‘MediaObjectSet’). *If ‘MediaObjectSet’’s external file is a GZIP archive:* Gives the media type of the GZIP archive member mapped to the ‘Object’.	string
start	A	NM/TM	0..1	*If ‘MediaObjectSet’’s external file is an uncompressed file, or else a GZIP archive containing one media object:* useless (implictly “true”). *If ‘MediaObjectSet’’s external file is a GZIP archive containing multiple media objects :* This attribute must be set to “true” for exactly one ‘Object’ and one only in the ‘Object’ sequence, the “start media object” on which the interactivity client must be launched. Default value, and applicable value for the other ‘Object’ elements : false	boolean
PartType	E4	NM/TM	0..N	Indicates the media types that should be supported also in order to correctly render an ‘Object’ consisting of several sub-media objects. E.g. a 3GP “Extended-presentation profile” would be one ‘Object’ with one “application/smil” ‘PartType’ advertising the presence of a SMIL presentation in the file.	string
File	E3	NO/TM	0..1	Present in case ALC without FLUTE is used for the delivery of ‘MediaObjectSet’’s external file. Structure identical to the ‘File’ child element of ‘FileDescription’ in the Access fragment. [BCAST10-SG].	
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-	A	NO/TM	0..1	See RFC 3926, section 3.4.2	unsignedByte

Encoding-ID					
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
SMSTemplate	E2	NM/TM	0..1	Contains the following attributes: relativePreference Contains the following elements: Description SelectChoice Note: the SMSTemplate is a media object set, although not encoded using the ‘MediaObjectSet’ generic structure. Note: The SMS Template provides information about the option(s) in an interaction, but does not contain rendering information. If rendering is to be specified by the service provider, the interaction can alternatively be described in an XHTML document with in-lined SMS URIs.	
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this media object set. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all the media object sets SHALL have mutually exclusive values of 'relativePreference'. If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all of these elements SHALL have the 'relativePreference' attribute instantiated. If only a single media object set is instantiated in 'MediaObjectGroup' then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt
Description	E3	NM/TM	0..N	Text describing the interaction to the end user, possibly in multiple languages. The language is expressed using the built-in XML attribute xml:lang with this element.	string

				This text can e.g. describe the overall scope of the interaction, valid for all interaction options described below. It might e.g. also contain information about the prize of the SMS interaction. For an interaction with only one choice (e.g. an offer to purchase merchandise like a ringtone), the 'Description' element SHOULD be used to provide information regarding the interaction and the 'ChoiceText' element MAY be discarded by the terminal.	
text	A	NO/TM	0..1	This attribute can contain a string that can be inserted into SMS messages specified by SMS-URI attributes below. Note: this attribute enables message size savings for the case where the same text appears in the SMS bodies of several choices, i.e. if multiple SelectChoice elements are present	string
SelectChoice	E3	NM/TM	1..N	Contains the following attributes: smsURI Contains the following elements: ChoiceText Note: For an interaction with multiple choices (like a voting between several options), the SelectChoice elements describe the different options to the user, and declare the SMS interaction to be executed when the user selects this option. For an interaction with one choice (e.g. an offer to purchase merchandise like a ringtone), there is only one SelectChoice element. Rendering of the choice(s) to the user is out of scope of this specification.	
smsURI	A	NM/TM	1	SMS receiver address and payload encoded as “sms:” URI scheme. Value of this attribute SHALL comply with “sms:” URI scheme [URI-Schemes], with the following exceptions: If the sms-body [URI-Schemes] of the sms URI scheme contains the string “$userid$”, it shall be replaced by the user ID. If the sms-body [URI-Schemes] of the sms URI scheme contains the string “$deviceid$”, it shall be replaced by the device ID. If the sms-body [URI-Schemes] of the sms URI scheme contains the string “$userinput$”, it should be replaced by a string that the user can enter. This may be an empty string. If $userinput$ is present in the SMS-URI, the terminal SHALL open the SMS template in SMS editor (or similar) to allow user input before sending the SMS. If, however, the $userinput$ string is not present in the sms-body, the terminal SHALL not provide the SMS for the end user to modify. The terminal SHOULD prompt the end user before sending the	anyURI

				SMS out. If the sms-body [URI-Schemes] of the sms URI scheme contains the string “$text$”, it SHALL be replaced by the string signalled in the attribute “Text” (if this attribute is present).	
ChoiceText	E4	NM/TM	0..N	Description of the interaction option, possibly in multiple languages. This is used to provide the end-user information on this interaction choice.. The language is expressed using the built-in XML attribute xml:lang with this element. For an interaction with one choice (e.g. an offer to purchase merchandise like a ringtone), the 'Description' element SHOULD be used to provide information regarding the interaction and the 'ChoiceText' element MAY be discarded by the terminal and the ChoiceText element MAY be omitted. For interactivity with multiple choices, the ´ChoiceText´ element SHALL be instantiated for each ´SelectChoice´.	string
EmailTemplate	E2	NO/TM	0..1	Contains attributes: relativePreference toHeader ccHeader bccHeader subjectHeader Contains the following elements: Description MessageBody Note: the EmailTemplate is a media object set, although not encoded using the ‘MediaObjectSet’ generic structure.	
relativePreference	A	NO/TM	0..1	This attribute gives the relative preference of this media object set. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all the media object sets SHALL have mutually exclusive values of 'relativePreference'. If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all of these elements SHALL have the 'relativePreference' attribute instantiated. If only a single media object set is instantiated in 'MediaObjectGroup' then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt
toHeader	A	NM/TM	1	The e-mail recipient(s) as defined in [RFC 2822]	string
ccHeader	A	NO/TM	0..1	The e-mail cc-recipient(s) as defined in [RFC 2822]	string
bccHeader	A	NO/TM	0..1	The e-mail bcc-recipient(s) as defined in [RFC 2822]	string

subjectHeader	A	NO/TM	0..1	The e-mail subject header as defined in [RFC 2822]	string
Description	E3	NO/TM	0..N	Description of the Email Template, possibly in multiple languages. This is used to provide the end-user extra information regarding the Email message. The language is expressed using the built-in XML attribute xml:lang with this element.	string
MessageBody	E3	NO/TM	0..1	The e-mail message body (text format defined in [RFC 2822] The value of this element SHALL be base64-encoded. Note: At least one of Subjectheader and MessageBody in an EmailTemplate SHOULD be present	base64Binary
VoiceCall	E2	NO/TM	0..1	Contains the following attributes: relativePreference Contains the following elements: Description PhoneNumber Note: the VoiceCallInteraction is a media object set, although not encoded using the 'MediaObjectSet' generic structure. It allows for voice call based interaction, by giving a description to the user and one or more telephone numbers that the user can call.	
relativePreference	A	NO/TM	0..1	This attribute gives the relative preference of this media object set. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all the media object sets SHALL have mutually exclusive values of 'relativePreference'. If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all of these elements SHALL have the 'relativePreference' attribute instantiated. If only a single media object set is instantiated in 'MediaObjectGroup' then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt
Description	E3	NM/TM	0..N	Text describing the interaction to the end user, possibly in multiple languages. The language is expressed using the built-in XML attribute xml:lang with this element. This text can e.g. describe the overall scope of the interaction, valid for all interaction options described below. It might e.g. also contain information about the prize of the voice call interaction.	string
PhoneNumbe	E3	NM/TM	1..N	Phone number to which the terminal initiates a	anyURI

r				voice call when the interactivity related to this InteractivityMediaDocument is triggered. The terminal SHALL prompt the user before actually making the call. If several phone numbers are present, the user SHALL be able to select the one to be used. A terminal with voice call capabilities MUST support telephone URI [RFC 3966]. Further, a terminal with SIP capabilities MUST support SIP URI [RFC 3261].	
Weblink	E2	NM/TM	0..1	This provides a reference to an external website. Contains attributes: - relativePreference - webURL Contains the following elements: - Description Note: the Weblink is a media object set, although not encoded using the 'MediaObjectSet' generic structure.	
relativePreference	A	NM/TM	0..1	This attribute gives the relative preference of this media object set. The greater value has higher priority to handle (i.e 2 has higher priority than 1). If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all the media object sets SHALL have mutually exclusive values of 'relativePreference'. If multiple media object sets are instantiated in this 'MediaObjectGroup ' then all of these elements SHALL have the 'relativePreference' attribute instantiated. If only a single media object set is instantiated in 'MediaObjectGroup' then the 'relativePreference' attribute MAY be instantiated for that element.	unsignedInt
webURL	A	NM/TM	1	URL to an external website.	anyURI
Description	E3	NM/TM	0..N	Description of the Weblink, possibly in multiple languages. This is used to provide the end-user extra information regarding the Weblink. The language is expressed using the built-in XML attribute xml:lang with this element.	string
AlternativeText	E2	NM/TM	0..N	Alternative Text to be displayed if none of the other media object sets is supported by the terminal Possibly in multiple languages. The language is expressed using the built-in XML attribute xml:lang with this element.	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	Any number of proprietary or application-specific elements that are not defined in this specification.	

				These elements may further contain sub-elements or attributes.	

Table 29: Data structure of InteractivityMediaDocument

5.3.6.1.3 On the rendering

The terminal SHALL render the information contained in the instances of ‘InteractivityMediaDocument’ when these are completely and successfully retrieved from the file delivery stream and when the interactivity is scheduled to take place, i.e. one or more InteractivityMediaDocuments are valid and are associated with the service or content that is being rendered at that moment. When instances of ‘InteractivityMediaDocuments’ with the same GroupID are valid at the same time, the terminal SHALL render those media objects in the document with the highest GroupPosition.

Upon parsing, or activation of, a received ‘InteractivityMediaDocument’ instance, the BCAST application SHALL identify the supported ‘MediaObjectSet’ instances according to:

- the interactivity technology(ies) supported, see section 5.3.6.1.4 below

AND

- supported language options for rendering the described interactivity, see section 5.3.6.1.6 below

and discard those instances that are not supported according to the two criteria above.

After having done this filtering step, the terminal obtains a list of languages supported for the interactive technologies it supports. From this list of languages, the terminal request the user to select one language, or perform this step automatically

Note: it is the responsibility of the network to ensure an instance of ‘InteractivityMediaDocument’ describes the interactivity in a given language for the given interactivity technology.

Furthermore the following applies:

- If multiple media object sets are instantiated in a 'MediaObjectGroup' the BCAST application SHALL render the media object set with the highest value of the 'relativePreference' attribute among the media object sets it supports.
- If only a single media object sets is instantiated in a 'MediaObjectGroup' the BCAST application SHALL render that media object if that media object set is supported.
- In the two previous cases, the BCAST application SHALL only select media object sets that correspond to the selected language or, alternatively, that apply to any language.
- If multiple 'MediaObjectGroups' are defined in the selected instance of 'InteractivityMediaDocument' the BCAST application SHALL go through all of them and render all the media object sets that are supported according to the three previous rules.
- The terminal SHALL support keeping track and rendering of several ‘InteractivityMediaDocument’ instances belonging to multiple groups (i.e. with different values of ‘groupID’) at the same time.

The InteractivityMediaDocument defines the actual details, which enable e.g. voting or ringtone ordering. The terminal SHALL be able to acquire and render the media objects attached to the ‘InteractivityMediaDocument’ without interrupting the acquisition and rendering of the ‘regular’ broadcast media stream.

5.3.6.1.4 MediaObjectSet parsing for interactivity technology selection

Information provided in the <MediaObjectSet> element is sufficient to determine whether the media object set is supported or not by the terminal. There is no need to open and parse the external file bundle. The terminal MAY take guidance of the following rules to determine this support :

- if <MediaObjectSet>'s external file is a single uncompressed file, the media object set SHOULD be seen as "supported" if :
    - the "Content-Type" attribute value of the <MediaObjectSet> is supported, and
    - if present, the 'PartType' s values of the 'Object' are all supported.
- if <MediaObjectSet>'s external file is an archive file, the media object set SHOULD be seen as "supported" if :
    - the "Content-Type" attribute value of each <Object> is supported, and
    - if present, the <PartType>s values in each <Object> are all supported.

5.3.6.1.5 InteractivityMediaDocument generation and parsing for language selection

The following table provides the list of elements that the terminal can use for language selection when parsing an instance of the <InteractivityMediaDocument>:

Element name	Language selection	Parent element
Description	Through the <xml:lang> attribute of this element	<SMSTemplate>, <EmailTemplate>, <VoiceCall>, and <Weblink> elements
ChoiceText	Through the <xml:lang> attribute of this element	<SelectChoice> element of the <SMSTemplate> element.
MediaObjectSet	Through the <xml:lang> attribute of this element	<MediaObjectGroup> element.

Table 30: elements of <InteractivityMediaDocument> used for language selection

In order for the terminal to provide a single language choice to the user (or perform an automatic selection), the language(s) available for a given interactivity have to be declared in a consistent manner across all the <MediaObjectGroup> instances in an <InteractivityMediaDocument> instance that describes such interactivity. In order to enable this, the server SHALL comply with the following rule:

- The instance of the <MediaObjectGroup> that has its <startMediaFlag> set to true SHALL explicitly declare all available languages for the interactivity scenario represented by the <InteractivityMediaDocument> instance, that is to say
    - If the said <MediaObjectGroup> provides any instance of <SMSTemplate>, <EmailTemplate>, <VoiceCall>, or <Weblink>, then the corresponding <Description> element SHALL be instantiated for each language
    - If the said <MediaObjectGroup> provides one or more instances of <MediaObjectSet>, there SHALL be at least one such instance per language option
- For each language declared as specified above
    - Each <Description> element as pointed by Table 30 that is to be used within its parent instance SHALL be instantiated for the said language
    - Each <ChoiceText> element as pointed the Table 30 that is to be used within its parent instance SHALL also be instantiated for the said language
    - For each instance of the <MediaObjectGroup> there SHALL be at least one instance of <MediaObjectSet> for the said language or, alternatively, an instance of <MediaObjectSet> configured for any language.
    - For any of the element pointed by Table 30 there SHALL NOT be any instance for a language that is not part of the available language options

Upon parsing the <InteractivitityMediaDocument> instance, the BCAST application identifies the available languages from the <MediaObjectGroup> instance that has the <startMediaFlag> set to true. The BCAST application MAY discard the languages that it does not support.

Upon activation of the <InteractivityMediaDocument>, the BCAST application SHALL select the media object sets for rendering that:

- are defined for the selected language or,
- are defined as applicable to any language

5.3.6.1.6 MediaObjectSet definition for some interactivity technologies

A media object set conveying an **MMS Message Template** conforming to [MMSTEMP] SHALL consist of the following:

- one GZIP archive file containing all the media objects (Message Template Definition, MMS presentation part, fixed/replaceable media objects).
- one <MediaObjectSet>, with Content-Type attribute set to “application/x-gzip”, and containing :
  - one “MTD” <Object>, with Content-Type attribute set to “application/vnd.omammsg-mtd+xml", and Start attribute set to “true”.
  - zero or one “MMS presentation part” <Object>, with Content-Type attribute set to “application/smil". If <MediaObjectSet> contains MMS presentation part, the sub-folder(s) SHALL NOT be used in <Content-Location> since MMS-SMIL cannot support sub-folder(s).
  - one <Object> per other bundled file, if any (fixed/replaceable media objects).

Note: If the end user decides to interact as triggered by Media Object Set of type **MMS Message Template**, it implies that the Terminal SHALL be able to execute any interaction over the Interaction channel by sending the MMS (the filled-in MMS Template).

A media object set conveying an **XHTML MP bundle** conforming to [XHTMLMP11] SHALL consist of the following:

- one GZIP archive file containing all the media objects (e.g. XHTML MP page(s), external ECMAScript MP files, external WAP CSS stylesheets, audio/visual media objects…).
- one <MediaObjectSet>, with Content-Type attribute set to “application/x-gzip”, and containing :
  - one “XHTML MP” <Object>, with Content-Type attribute set to “application/vnd.wap.xhtml+xml" and Start attribute set to “true”.
  - one <Object> per other bundled file, if any (that may be additional XHTML MP pages).

Note**:** If the end user decides to interact as triggered by Media Object Set of type **XHTML MP bundle**, it implies that the Terminal SHALL be able to execute any interaction over the Interaction channel by executing HTTP requests (following the hyperlinks present in XHTML). Further, if the Terminal supports SMS-based messaging, the Terminal SHALL be able to support “sms:”-URI scheme as defined in section 5.3.6.1 and consequently be able to perform SMS-based interaction over the Interaction channel.

A media object set conveying a **3GPP PSS SMIL bundle** conforming for the presentation part to [3GPP 26.246R6]) SHALL consist of the following:

- one GZIP archive file containing all the media objects (SMIL presentation, audio/visual media objects…).
- one <MediaObjectSet>, with Content-Type attribute set to “application/x-gzip”, and containing :
  - one “3GPP PSS SMIL” <Object>, with Content-Type attribute set to “application/smil" and Start attribute set to “true”.
  - one <Object> per other bundled file, if any.

Note: If the end user decides to interact as triggered by Media Object Set of type **3GPP PSS SMIL bundle**, it implies that Terminal SHALL be able to execute any interaction over the Interaction channel by executing HTTP requests (following the hyperlinks present in SMIL). Further, if the Terminal supports SMS-based messaging, the Terminal SHALL be able to support “sms:”-URI scheme as defined in section 5.3.6.1 and consequently be able to perform SMS-based interaction over the Interaction channel.

A media object set conveying a **3GPP2 MSS SMIL bundle** conforming for the presentation part to [3GPP2 C.S0050]) SHALL consist of the following:

- one GZIP archive file containing all the media objects (SMIL presentation, audio/visual media objects…).
- one <MediaObjectSet>, with Content-Type attribute set to “application/x-gzip”, and containing :
    - one “3GPP2 MSS SMIL” <Object>, with Content-Type attribute set to “application/smil" and Start attribute set to “true”.
    - one <Object> per other bundled file, if any.

Note: If the end user decides to interact as triggered by Media Object Set of type 3GPP2 MSS SMIL bundle, it implies that Terminal SHALL be able to execute any interaction over the Interactive Channel by executing HTTP requests (following the hyperlinks present in SMIL). Further, if the Terminal supports SMS-based messaging, the Terminal SHALL be able to support “sms:”-URI scheme as defined in section 5.3.6.1 and consequently be able to perform SMS-based interaction over the Interactive Channel.

See Appendix C for some informative examples of <MediaObjectSet> elements.

5.3.6.1.7 Using URI scheme “sms:”

Terminals that support SMS-based messaging and/or that support XHTML based Media Object Sets SHALL support the “sms:” URI scheme as specified in [URI-Schemes] as a valid scheme for hyperlinks.

5.3.6.1.8 Service Interaction using MMS Message Template

This section describes how to retrieve and use MMS Message Template for Service Interaction.

The terminal SHALL retrieve MMS Message Template from InteractivityMediaDocument (refer to 5.3.6.1). The terminal MAY retrieve MMS Message Template from MMS.

The terminal SHOULD store MMS Message Templates in its storage area after retrieval.

The terminal MAY use Application ID described in [MMSCONF] to launch client software, which handles MMS Message Template (MMS Message Template Client), in the case that the Template is retrieved from MMS.

5.3.7 Service Interaction launch and feedback

The terminal SHALL launch MMS Message Template Client according to the timing described in InteractivityMediaDocument , in a similar way to the other Service Interaction methods.

MMS Message Template Client SHALL create Multimedia Message (MM) according to the process defined in MMS Message Template Definition (MTD) [MMSTEMP].

After creating the resulting MM, MMS Message Template SHOULD send the Message to Service Application address defined in MTD.

5.3.7.1 Broadcast delivery of InteractivityMediaDocuments

The broadcast delivery of the instances ‘InteractivityMediaDocument’ and any associate files has the following characteristics and constraints. For the delivery the network SHALL

- use FLUTE file delivery session containing at least one FDT Instance,

- list all the delivered files in every instance of FDT,
- use the string “application/vnd.oma.bcast.imd+xml” as the value of ‘Content-Type’ for every instance of ‘InteractivityMediaDocument’ in every FDT Instance and
- use the following convention for allocating the ‘Content-Location’ values for the instances of the ‘InteractivityMediaDocument’: <GroupID>:<GroupPosition> where the
- <GroupID> stands for the group identifier and <GroupPosition> for the group position represented by the instance of ‘InteractivityMediaDocument’.

Furthermore in the case of broadcast delivery the network SHALL use the string “oma:bcast1.0:imd:” as the prefix of the interactivity media group identifier (GroupID).

5.3.7.2 Interactive delivery of InteractivityMediaDocuments

5.3.7.2.1 Transport protocols

There are the two following mechanisms for delivering InteractivityMediaDocuments to the terminal using the interactive channel:

- using HTTP as the transport the terminal specifically requesting the InteractivityMediaDocuments from the network and
- using OMA PUSH the network pushing the InteractivityMediaObjects to the terminals.

If the terminal supports the interaction channel, the terminal SHALL support the former and additionally if the terminal supports OMA PUSH, the terminal SHALL also support the latter.

When the InteractivityMediaDocuments are delivered using OMA PUSH the content type SHALL be set to “application/vnd.oma.bcast.imd+xml”.

5.3.7.2.2 InteractivityMediaDocument request messages

When the terminal requests InteractivityMediaDocuments from the network, the terminal SHALL use HTTP POST

with the following syntax : “POST <interactivityMediaURL> HTTP/1.1\r\n<InteractivityMediaDocumentRequest>” where <interactivityMediaURL> denotes the destination for the HTTP requests as signaled in the ‘interactivityMediaURL’ attribute of the ‘InteractivityData’ fragment representing the interactivity in question, see section 5.1.2.10 of [BCAST10-SG].. Both the HTTP POST request and the corresponding HTTP response SHALL also contain the following HTTP header fields:

- ‘Content-Length’,
- for request message: ‘Content-Type’ which SHALL be set to “text/xml”.
- for response message: ‘Content-Type’ which SHALL be set to “multipart/mixed” and
- ‘Host’ in case the ‘Request-URI’ is not in the absolute form specified in [RFC 2616].

The XML structure in Table 31 defines the syntax for the ‘InteractivityMediaDocumentRequest’ placed into the payload of the HTTP POST request.

The HTTP response of the HTTP POST request response message SHALL be of type “multipart/mixed”.

The first body part of the multipart in the response:

- SHALL contain one ‘InteractivityMediaDocumentResponse’ XML document as defined in Table 32.
- SHALL include Content-Type header set to ‘text/xml’

Other body parts may follow the first body part in the response. In that case each body part:

- SHALL contain one file representing the full set of media objects associated to exactly one <MediaObjectSet> of a MediaObjectGroup of the returned InteractivityMediaDocument. This file SHALL be either one uncompressed media file (e.g. 3GP file) being the media object itself, or one GZIP archive file containing the compressed media objects, as described in section 5.3.6.1.2.

- SHALL include Content-Location header set to Content-Location attribute value of <MediaObjectSet> element.
- SHALL include Content-Type header, set to actual MIME type of uncompressed media file (e.g. ‘video/3gpp’) or to ‘application/x-gzip’if the media objects are carried in a GZIP archive.

In case the response message does not contain all the files associated with the 'InteractivityMediaDocuments' contained in the response message, the terminal MAY use HTTP GET to retrieve these missing files.

Name	Type	Category	Cardinality	Description	Data Type
InteractivityMediaDocumentRequest	E			The request to be used by the terminal to request InteractivityMediaDocuments. Contains the following attributes: requestID Contains the following elements: UserID DeviceID GroupID	
requestID	A	O	0..1	Identifier for the InteractivityMediaDocument request message.	unsignedInt
UserID	E1	O	0..N	The user identity known to the BSM. Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
DeviceID	E1	O	0..N	A unique device identification known to the BSM. Contains the following attributes: type	string
type	A	M	1	Specifies the type of Device ID. Allowed values are 0 – reserved for future use 1 – IMEI [3GPP TS 23.003] 2 – MEID [3GPP2 C.S0072] 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
GroupID	E1	M	1	ID of the requested group of InteractivityMediaDocument, globally unique The GroupID is carried in BCAST SG fragment called InteractivityData	anyURI

Table 31: Structure of Interactivity Media Document Request

Name	Type	Category	Cardinality	Description	Data Type
InteractivityMediaDocumentResponse	E			The response to the 'InteractivityMediaDocumentRequest' message. Contains the following attributes: requestID statusCode Contains the following elements: InteractivityMediaDocument	
requestID	A	O	0..1	Identifier for the corresponding InteractivityMediaDocument request message.	unsignedInt
status Code	A	M	1	The overall outcome of the request, according to the return codes defined in section 5.11.	unsignedByte
InteractivityMediaDocument	E1	M	1	The InteractivityMediaDocument as specified in 5.2.6.1	complexType

Table 32: Structure of Interactivity Media Document Response

5.4 Personalization/Support for User-based Profiles and Preferences

5.4.1 User-based Profiles over Broadcast Channel

The BCAST Enabler enables targeted reception through delivery of user-based profiles over the broadcast channel using the Service Guide. The "TargetUserProfile" element of Service Guide SHALL be used for that purpose.

Exact terminal behavior for interpreting the "TargetUserProfile" is not specified. However, the terminal MAY be able to filter the Service Guide based on the "TargetUserProfile".

5.4.2 Communicating the End User Preferences to Network

The terminal MAY communicate the End User preferences to the network using the scheme defined in this section. Both the Terminal and the network MAY support the scheme. The behavior of the network and any subsequent actions beyond providing the End User preferences are not specified in BCAST Enabler.

The data structure for communicating the End User preferences from terminal to network is as follows:

Name	Type	Category	Cardinality	Description	Data Type
EndUserPreferences	E	O		The end user preferences signalled to the Service Provider Contains the following elements: UserID Preference	
UserID	E1	M	1	User Identity known to the BSM. It describes The identification of the end user whose preferences are described here. Contains the following attribute:	string

				type	
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
Preference	E1	M	1..N	The attribute-value pair describing an individual preference. NOTE: the exact attribute for preference shall be defined by service or content provider. Contains the following attributes: attribute value	
attribute	A	M	1	Attribute being described	string
value	A	M	1	Value of the attribute	string

Table 33: Structure of End User Preference Message

The above data structure SHALL be instantiated as XML instance according to XML Schema [BCAST10-XMLSchema-Userpreference].. The XML instance in turn SHALL be communicated from terminal to network by HTTP POST. For confidentiality, HTTPS based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246] MAY be used.

5.5 Charging

This section specifies the use of OMA Charging Enabler to realize the charging of OMA Mobile Broadcast Services. OMA Charging Enabler defines a set of interfaces to allow other Enablers to access the charging functionality. The interfaces are specified in [OMA Charging AD]. This section defines how, when and by whom the charging is triggered and which functional entity invokes the charging using the interface of OMA Charging Enabler. This section also defines the data that will be exchanged within the charging event.

5.5.1 Chargeable Events in the Scope of the BCAST Enabler

Chargeable event is a service related event that has taken place, and can be specified and recorded. This section identifies chargeable events in the scope of the OMA Mobile Broadcast Services technical specification. It should be noted that chargeable events can also occur for example in a Broadcast Distribution System or in other entities of the OMA BCAST Architecture to record the usage of the mechanisms that they provide (e.g. distribution and protection mechanisms) but these chargeable events are not specified in this document.

Not all chargeable events lead necessarily to a *charging event*, i.e. the sending of charging information to the Charging Enabler for further processing. The events that are actually charged for can depend on the implementation. Therefore, the list in this section should be regarded as a list of events that potentially trigger charging events.

Chargeable Event	Section where defined	Source of the event
Subscription-Based Charging		
Subscribe/Purchase Request	5.1.5, 5.1.6	BSM
End-user subscribes or purchases a certain service based on information received through the Service Guide.	[BCAST10-Architecture] 5.4.6.1	
Subscription Update	5.1.5, 5.1.6	BSM
In case of open-ended subscriptions, the BSM may need to generate	[BCAST10-Architecture]	

charging information from time to time until the subscription is cancelled.	5.4.6.7	
Unsubscribe Request Open-ended subscriptions, and possibly other subscriptions, are valid until they are cancelled by the end-user. Depending on the contract, they may also have to be cancelled (and renewed by issuing a new order request) when the price per subscription period changes.	5.1.6.7 [BCAST10-Architecture] 5.4.6.8	BSM
Consumption-Based Charging		
Token Purchase Request Token Purchase Request can be used to order tokens that can be used in consumption-based charging models. As to calls to the Charging Enabler, tokens can be used in two ways: • Pre-paid tokens: When the BCAST client orders tokens, BSM calls the Charging Enabler and tokens are charged as they are ordered before the actual service delivery • Post-paid tokens: When the BCAST client orders tokens, if the subscriber uses online charging, a respective credit reservation is made. In the offline case, a positive credit response is assumed implicitly. Used service units are reported to the Charging Enabler only when the BCAST client reports used tokens to the BSM. **NOTE!** It is important to note here that the prepaid/postpaid distinction is independent of the type of the subscriber's account in the Charging Infrastructure (i.e. pre-paid or post-paid subscription).	5.1.5, 5.1.6 [BCAST10-Architecture] 5.4.6.9	BSM
Service Interaction		
Interactive Service Ordering The end-user reacts to an interaction pointer and requests for an additional service, such as voting or related value-added content. Charging for interactive service ordering is in the BCAST Enabler's scope only in simple cases where the additional service can be identified with a simple combination of a purchase item ID and purchase option or equivalent. In more complex cases, it is likely that service interaction is redirected to a separate application the charging of which is outside the scope the BCAST Enabler.	[BCAST10-Architecture] 5.4.5	BSI-G

Table 34: List of chargeable events

5.5.2 When to Trigger Calls to the Charging Enabler

This section identifies when charging information needs to be sent to the Charging Enabler in relation to the different chargeable events.

In the case of Subscription/Purchase Request, Subscription Update, Unsubscribe, Token Purchase Request, or Interactive Service Ordering, the high-level charging flow is the following:

- When the request arrives, before service delivery
  - The BCAST Enabler implementation may know based on pre-configured information or through a query to an external system whether online or offline charging interface should be used towards the Charging Enabler. It this information is not available, the BCAST Enabler may assume online and make the first request to the online (CH-2) interface, which may return an error code indicating that offline should be used.
  - If online charging is to be used, send an Initial Request using CH-2 to make a credit reservation
- During service delivery

  - In the online case, Interim Requests to CH-2 may be needed if the quota granted in the previous step(s) is depleted

- After service delivery
  - If the online-offline determination outcome was offline, report service usage using CH-1
  - If the online-offline determination outcome was online, report the final service usage step using Termination Request of CH-2

5.5.3 BCAST-related Information in Charging Messages

This section specifies how charging information for BCAST services is mapped to OMA Charging Data Elements of the Charging Enabler.

5.5.3.1 Subscription-Based Charging: Subscribe/Purchase, Subscription Update, Unsubscribe Request

BCAST field name or value constants	Type	OMA Data Elements in Charging interface	Description
Value: BCAST@openmobilealliance.org	String	Service Context Id	Fixed value to identify the service specification in the context of which the charging events must be interpreted.
Values: SUBSCRIBE, SUBSCRPITION_UPDATE, UNSUBSCRIBE (for Subscription-Based Charging)	String	Service Identifier	Identifies more precisely the type of service within the context defined by the Service Context Id. NOTE: Different from Service ID.
Field: UserID	String	Subscription Id Data	The globally unique identity of the subscriber
Field: type attribute under UserID	unsignedByte	Subscription Id Type	Type of the subscriber identity (e.g. MSISDN, IMSI, SIP_URI)
Field: PurchaseItemID	anyURI	Service Key	The globally unique ID of the Service Guide fragment that describes what the end-user has ordered or cancelled. It should be noted that a particular Service Item may be available through several Purchase Items (e.g. because of bundling and several order options or purchase channels).
Values: depending on context	String	Correlation Id	Depending on the deployment, different identifiers can be used here to enable correlation between the charging events generated by BCAST service entities and charging events generated by other entities (such as distribution entities or content protection mechanisms).
Field: Price	decimal	Unit Value, Value Digits, Exponent	Amount to be reserved/debited from the end-user’s account. In case of reservation, the listed data elements must be included in the requested service units data element. In case of reporting units to be debited, the used service units data element must be used in the

			charging interface.
Field: currency	String	Currency Code	Numeric representation of Currency Code as specified in ISO4217
Field: DeviceID	String	User Equipment Info Value	A unique device identification known to the BSM
Field: type attribute under DeviceID element	unsignedByte	User Equipment Info Type	The type of the unique device identification (e.g. IMEI, MEID).

Table 35: Mapping table for Subscription based Charging

5.5.3.2 Consumption-Based Charging: Token Purchase Request

BCAST field name or value constants	Type	OMA Data Elements in Charging interface	Description
Value: BCAST@openmobilealliance.org	String	Service Context Id	Fixed value to identify the service specification in the context of which the charging events must be interpreted.
Values:TOKEN_PURCHASE (for Consumption-based charging)	String	Service Identifier	Identifies more precisely the type of service within the context defined by the Service Context Id. NOTE: Different from Service ID.
Field: UserID	String	Subscription Id Data	The globally unique identity of the subscriber
Field: type attribute under UserID	unsignedByte	Subscription Id Type	Type of the subscriber identity (e.g. MSISDN)
Field: PurchaseItemID	anyURI	Service Key	The globally unique ID of the Service Guide fragment that represents the token product.
Values: depending on context	String	Correlation Id	Depending on the deployment, different identifiers can be used here to enable correlation between the charging events generated by BCAST service entities and charging events generated by other entities (such as distribution entities or content protection mechanisms).
Field: currency	String	Currency Code	Numeric representation of Currency Code as specified in ISO4217. If the Currency Code element is present, the Unit Value, Value Digits and Exponent elements below will be used. If the CurrencyCode element is not given, the Service Specific Units element below will be used.
Field: Price	decimal	Unit Value, Value Digits, Exponent	Amount to be reserved/debited from the end-user's account. These sub-elements of the Money data element are used in the charging interface if the BCAST Enabler is able to determine the price of the request (either in monetary or non-

			monetary terms). In case of reservation for post-paid tokens, the listed data elements must be included in the requested service units data element. In case of reporting used post-paid tokens or ordering pre-paid tokens, the used service units data element must be used in the charging interface.
Field: Price	decimal	Service Specific Units	Amount of tokens to be reserved/debited from the end-user's account. The Service specific units data element is used in the charging interface if price determination is left to the Charging Enabler. In case of reservation for post-paid tokens, the listed data elements must be included in the requested service units data element. In case of reporting used post-paid tokens or ordering pre-paid tokens, the used service units data element must be used in the charging interface.
Field: DeviceID	String	User Equipment Info Value	A unique device identification known to the BSM
Field: type attribute under DeviceID element	unsignedByte	User Equipment Info Type	The type of the unique device identification (e.g. IMEI, MEID)

Table 36: Mapping table for Consumption based Charging

5.5.3.3 Service Interaction

Service interaction pointers may lead the end-user to a completely different service from BCAST (e.g. to MMS sending), and these external services usually have their own charging which is not in the scope of this specification. This specification, however, caters for cases where the additional interactive service does not have charging specified separately and the price of the interaction transaction is available to the BCAST Enabler or some part of the BCAST Enabler implementation can determine the price. Also cases where price determination is delegated to the Charging Enabler but price can be calculated simply based on the InteractivityDataId accessed can be supported.

BCAST field name or value constants	Type	OMA Data Elements in Charging interface	Description
Value: BCAST@openmobilealliance.org	string	Service Context Id	Fixed value to identify the service specification in the context of which the charging events must be interpreted.
Value:SERVICE_INTERACTION (for Service Interaction)	string	Service Identifier	Identifies more precisely the type of service within the context defined by the Service Context Id. NOTE: Different from Service ID.
Field: UserID	string	Subscription Id Data	The globally unique identity of the subscriber
Field: type attribute under UserID	unsignedByte	Subscription Id Type	Type of the subscriber identity (e.g. MSISDN)

Field: InteractivityDataID	anyURI	Service Key	The globally unique ID of the Service Guide fragment that describes what the end-user has accessed.
Values: depending on context	string	Correlation Id	Depending on the deployment, different identifiers can be used here to enable correlation between the charging events generated by BCAST service entities and charging events generated by other entities (such as distribution entities or content protection mechanisms).
Field: Price	decimal	Unit Value, Value Digits, Exponent	Amount to be reserved/debited from the end-user's account. In case of reservation, the listed data elements must be included in the requested service units data element. In case of reporting units to be debited, the used service units data element must be used in the charging interface.
Field: currency	string	Currency Code	Numeric representation of Currency Code as specified in ISO4217
Field: DeviceID	String	User Equipment Info Value	A unique device identification known to the BSM
Field: type attribute under DeviceID element	unsignedByte	User Equipment Info Type	The type of the unique device identification (e.g. IMEI, MEID)

Table 37: Mapping table for Service Interaction

5.5.4 Exchange of charging data among systems

It can be assumed that entities that are reflected in the BCAST architecture may need to exchange business related data.

However, the BCAST enabler does not specify a defined format for the exchange of charging data between Broadcast Service Providers, or between a Broadcast Service Provider and a Content Provider.

5.6 Mobility

The location of the Terminal may change over time. Different usage scenarios typically involve different rates of change in the location of the Terminal. However, what is significant in the change is not the speed of the change but the fact that the change in the location of Terminal may involve a change in the set of available Mobile Broadcast Services. Along with the change in the location of Terminal the currently available transmission may become unavailable due to changing radio reception conditions. Alternatively, the change in Terminal's location may move the Terminal away from its currently available Broadcast Service Area. In both cases the current set of available Mobile Broadcast Services may change.

There are two cases to consider in the context of mobility and Mobile Broadcast Services. Firstly, the terminal may be currently receiving a Mobile Broadcast Service which is affected by the change. Secondly, the terminal may only be receiving and updating the Service Guide that is related to the Service, affected by the change. Both cases are exceptions in a normal service consumption process and require handling. In the former case, the change affects the current access to the Service while in the latter case the change affect to the possible ways of accessing the Service Guide.

This section provides normative specification for the network side (Service Guide function) to support the mitigation of mobility effects. On the network side the support for broadcast mobility is centralized in the Service Guide function. The methods outlined in the following sections are supported by the SG-D and MAY be used in the transmitted Service Guide.

5.6.1 Specifying Alternative Accesses for a Service

Service Guide allows describing several Accesses for a particular Service. The Service Guide can declare a Service in the Service Guide that MAY have several Accesses associated with it. In case the selected Access becomes unavailable due to mobility (or some other reason), the Terminal MAY continue accessing the Service via another Access given that the other Access semantically represents same or similar component of the Service.

5.6.2 Global Identification of Services and Content

The Service Guide MAY declare global identification for both Service (attribute GlobalServiceID in Service Fragment) and Content (attribute GlobalContentID in Content Fragment). Two fragments with the same global identifiers describe the same asset. How the terminal uses the global service identifier or the global content identifier is out of scope of this specification.

5.7 Broadcast Roaming

Broadcast Roaming allows a user to receive Broadcast Services from a Mobile Broadcast Service Provider different from his Home Mobile Broadcast Service Provider. This can happen, for example, when the user is not able to access the services provided by Home Mobile Broadcast Service Provider. In that case the Broadcast Roaming enables the user to receive Broadcast Services from another Mobile Broadcast Service Provider independent on the underlying Broadcast Distribution System.

The Mobile Broadcast Services (BCAST) 1.0 Enabler enables the Broadcast Roaming through the use of various functions of the enabler: through the Service Guide, through roaming signaling between Terminal and Visited Mobile Broadcast Service Provider, through roaming signaling between Visited Mobile Broadcast Service Provider and Home Mobile Broadcast Service Provider and through the Terminal Provisioning function. The following gives the overview on how these functions relate in the context of Broadcast Roaming:

- Service Guide Delivery Descriptors (SGDD) within the Service Guide declare the existence and the availability of Service Guide fragments. The SGDD allows the Terminal to deduce which fragments are associated with which Mobile Broadcast Service Provider (through use of BSMFilterCodes). Related to this signaling, there are visibility rules that the terminals are expected to comply with. Further, SGDD enables a method to convey points of contact which the visiting terminals can contact in case Broadcast Roaming is needed. This aspect of Broadcast Roaming is normatively specified within the specification of SGDD, in section 5.4.1.5 of [BCAST10-SG].
- Terminal Provisioning enables the Home Mobile Broadcast Service Provider to maintain a terminal-resident elements used by the roaming function. These elements include the list of Mobile Broadcast Service Providers (their BSMFilterCodes) affiliated with the terminal as well as entry details of default roaming contact point - the server that terminal can send roaming requests in the case terminal does not find any other entry points within the Service Guide signaling. They also include parameter that determines whether the terminal initiates the service provisioning requests to Visited BSM or to Home BSM. Finally, these elements include parameters that can be used to control terminal behavior in the context of Broadcast Roaming: an element that controls whether roaming requests should always be sent to Home BSM and an element that determines terminal behavior for fragments that are not associated with any BSMSelectors. These aspects of Broadcast Roaming are normatively specified within this document, Appendix F (Management Object). In addition to using Terminal Provisioning, the management information in Appendix F can be pre-configured in the Terminal, or can be conveyed to the terminal by some other means which are out of scope of this specification.
- Roaming Rule request and response messages between Terminal and BSM associated with Home and/or Visited Mobile Broadcast Service Provider allow Terminals to request and Mobile Broadcast Service Providers to provide the visibility constraints defined by Roaming Rules. This aspect of Broadcast Roaming is normatively specified within this document (section 5.7.1). The contact points for the request messages are signaled within the SGDDs – that aspect of Broadcast Roaming is normatively specified within the specification of SGDD, in section 5.4.1.5 of [BCAST10-SG].

- Specific Service Provisioning messages that enable Terminal to request for service, request for Tokens and request for renewal of subscriptions. In the context of Broadcast Roaming, the Service Provisioning messages sent by the Terminal trigger roaming message exchange between Home and Visited Mobile Broadcast Service Provider. This aspect is normatively specified within this document (section 5.1). Subsequent of successful Roaming Service Response, LTKMs can be delivered to the terminal (via Push LTKM with Smartcard profile or Trigger with DRM profile). The LTKM acquisition is not covered in this document as it is a Service and Content protection procedure.
- The roaming messages between Home and Visited Mobile Broadcast Service Providers allow the either the Home or Visited Mobile Broadcast Service Provider to initiate the roaming as a reaction to initial user roaming request. This aspect of Broadcast Roaming is normatively specified within this document (section 5.7.2).
- The informative walk-through of Broadcast Roaming is given in this document (Appendix E).

Broadcast Roaming in BCAST 1.0 allows a Terminal to be associated with multiple Home BSMs (and hence multiple BSMFilterCodes). While this allows a model wherein the Terminal is associated with different service providers, the primary use of this functionality will be of specifying different subscription types per a single provider.

Roaming agreements between Home Mobile Broadcast Service Provider and Visited Mobile Broadcast Service Provider and the related trust relationship are out of BCAST scope.

Both the Network and the Terminal MAY support Broadcast Roaming. If the Network supports Broadcast Roaming, backend interfaces for roaming SHALL also be supported.

Note: playback of protected content recorded while roaming may have limitations due to the inability of the terminal to retrieve rights once back in its home network. Such use case may not be supported until a later release of the present specification.

5.7.1 Roaming messages between Terminal and BSM

Terminal uses the RoamingRuleRequest to request the RoamingRules associated with BSMSelector (identified by the id of the selector). As a response, the Terminal receives RoamingRuleResponse that carry the RoamingRules.

The XML schema for these messages is defined in [BCAST10-XMLSchema-Roaming-frontend].

5.7.1.1 RoamingRuleRequest

Name	Type	Category	Cardinality	Description	Data Type
RoamingRuleRequest	E			Request message of Roaming Rules. Contains the following elements: UserID RequestEntry	
UserID	E1	M	1	A unique ID that SHALL be used to identify the terminal in BCAST service area of both the Home Mobile Broadcast Service Provider and Visited Mobile Broadcast Service Provider. Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are:	unsignedByte

				0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	
RequestEntry	E1	M	1..N	This element represents a request for roaming rules associated with the given ‘HomeBSMFilterCode’ and ‘VisitedBSMFilterCode’ instances. It is expected that the terminal will request rules separately (i.e. with different ‘RequestEntry’ instances) for each Visited BSM of interest. However, in case the terminal provides more than one instance of ‘VisitedBSMFilterCode’ in a ‘RequestEntry’ instance, the server SHALL assume that the ‘HomeBSMFilterCode’ instance applies for all the given ‘VisitedBSMFilterCode’ of the same ‘RequestEntry’ instance. As of BCAST 1.0, the terminal SHALL provide one and only one instance of the ‘RequestEntry’ element in a given ‘RoamingRuleRequest’ message. That instance MAY include an instance of the child ‘HomeBSMFilterCode’ element that, when instantiated, SHALL represent the Home BSM the terminal is affiliated with. Contains the following elements: HomeBSMFilterCode VisitedBSMFilterCode	
HomeBSMFilterCode	E2	M	0..1	The code that specifies the Home BSM the terminal is affiliated with. This element has the same structure as the ‘BSMFilterCode’ element in the ‘ServiceGuideDeliveryDescriptor’.	complexType as defined for ‘BSMFilterCode’ in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSMFilterCode	E2	M	1..N	The code that specifies the Visited BSM. This element has the same structure as the ‘BSMFilterCode’ element in the ‘ServiceGuideDeliveryDescriptor’.	complexType as defined for ‘BSMFilterCode’ in section 5.4.1.5.2 of [BCAST10-SG]

Table 38: Structure of RoamingRuleRequest Message

5.7.1.2 RoamingRuleResponse

Name	Type	Category	Cardinality	Description	Data Type
RoamingRuleResponse	E			Response message of Roaming Rules Contains the following attribute: globalStatusCode Contains the following element: ResponseEntry	
global Status Code	A	M	0..1	The overall outcome of the request, according to the return codes defined in section 5.11. This attribute also governs the way the 'itemwiseStatusCode' attribute is instantiated in this response: ▪ If this attribute is present and set to value "0", the request was completed successfully. In this case the 'itemwiseStatusCode' SHALL NOT be given per each 'ResponseEntry' ▪ If this attribute is present and set to some other value than "0", there was a generic error concerning the entire request. In this case the 'itemwiseStatusCode' SHALL NOT be given per each requested 'ResponseEntry' ▪ If this attribute is not present, there was an error concerning one or more 'RequestEntry' elements associated with the request. Further, the 'itemwiseStatusCode' SHALL be given per each 'ResponseEntry'.	unsignedByte
ResponseEntry	E1	M	0..N	Entry containing response to each requested Visited BSM. This element SHALL be instantiated in case of successful completion of the related request. Contains the following attribute: itemwiseStatusCode exclusive Contains the following element: VisitedBSMFilterCode HomeBSMFilterCode RoamingRule	
itemwiseStatusCode	A	M	0..1	Specifies a status code of each PurchaseItems using GlobalStatusCode defined in the section 5.11.	unsignedByte
exclusive	A	O	0..1	Indicates whether the rules delivered in this response are exclusive when executed. If "true", the rules are exclusive and terminal that accesses fragments covered by these rules (i.e. associated with the BSMFilterCode declared in the 'VisitedBSMFilterCode' instance) SHALL NOT access fragments associated with any other BSM. This means that – if the terminal selects a Visited BSM for which the rule are marked with the value	boolean

				"true" of this attribute the Terminal SHALL only use the SG fragments of the selected BSM and not mix SG fragments from other BSM even if the Terminal already got access to those. If "false", the rules are not exclusive and, upon selection of the related Visited BSM, the terminal can access fragments associated with any other BSM. Default value of this attribute is "false".	
VisitedBSMFilterCode	E2	M	1	The code that specifies the Visited BSM for which this ResponseEntry applies. This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'BSMFilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
HomeBSMFilterCode	E2	M	0..N	The code that specifies the Home BSM for which this ResponseEntry applies. This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'. Note that a RoamingRule can apply to several Home BSM for a given Visited BSM. In case no HomeBSMFilterCode instance is given, the terminal SHALL interpret the associated RoamingRule as applicable to any Home BSM. In case the BSM instantiates this element, it SHALL provide one instance corresponding to the 'HomeBSMFilterCode' provided by the terminal in the related 'RoamingRuleRequest' message.	complexType as defined for 'BSMFilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
RoamingRule	E2	M	1..N	Entry specifying the RoamingRule between the Visited BSM and Home BSM declared in the parent 'ResponseEntry' instance. This element has the same structure as the 'RoamingRule' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'RoamingRule' in section 5.4.1.5.2 of [BCAST10-SG]

Table 39: Structure of Roaming RuleResponse Message

In case a roaming rule is not specific to any given Home BSM, it is RECOMMENDED that:

- in case the roaming rule is not subject to frequent changes, the Network delivers it following a RoamingRuleRequest from the terminal in a RoamingRuleResponse.
- and, in case the roaming rule is subject to frequent changes, the Network delivers it through the 'RoamingRule' element in the SGDD.

Note: delivery of roaming rules through SGDD over the interaction channel is not subject to either any recommendations or limitations with regard to the considerations defined above.

However, in case a roaming rule is specific to a given Home BSM, it SHOULD NOT be delivered via the 'RoamingRule' element of the SGDD.

5.7.1.3 Transport protocol

The BSM and Terminal SHALL support HTTP 1.1 [RFC 2616] as a delivery method to exchange roaming messages. The BSM and Terminal MAY also support HTTPS for this purpose, where HTTPS SHALL be based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246]. Furthermore, the following rules apply:

- The Terminal SHALL issue HTTP/1.1 POST request carrying one 'RoamingRuleRequest' element in the 'message-body' part.
- The BSM SHALL answer with an HTTP/1.1 200 (OK) response carrying one, and only one, 'RoamingRuleResponse' element in the 'message-body' part
- In both request and responses, the Content-Type entity-header field SHALL be set to 'application/vnd.oma.bcast.roaming+xml'
- The BSM MAY compress the response using GNU zip [GZIP], in which case the Content-Encoding entity-header field SHALL be set to 'gzip'.

5.7.2 Roaming messages between Home BSM and Visited BSM

Roaming messages between Home BSM and Visited BSM are used to carry out the roaming negotiation between the two BSMs. The exchange of these messages is triggered by the Terminal sending the Service Provisioning message. Four cases exist as follows.

If the value of Management Object "<X>/Roaming/UseVisitedServiceProvisioningMode" is assigned with value "false" the following SHALL apply:

- Terminal sends Home BSM the Service Request message involving service provided by the Visited BSM. If the Home BSM deduces from the message that it needs to contact Visited BSM to get clearance for the request, the Home BSM SHALL send the 'RoamingServiceRequest' (section 5.7.2.2) to the Visited BSM. Visited BSM SHALL respond to the request by sending 'RoamingServiceResponse' (section 5.7.2.3).). In case the response allows roaming, then the Home BSM sends a successful 'ServiceResponse' to the terminal. This leads to a subsequent LTKM delivery (push LTKM with Smartcard Profile or Trigger with DRM Profile).

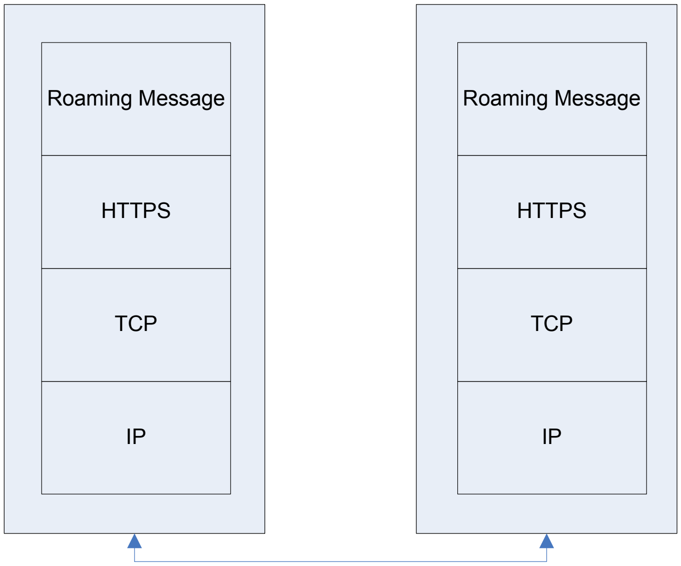
If the value of Management Object "<X>/Roaming/UseVisitedServiceProvisioningMode" is assigned with value "true" the following SHALL apply:

- Terminal sends Visited BSM the Service Request message involving service provided by the Visited BSM. If the Visited BSM deduces from the message that it needs to contact Home BSM to get clearance for the request, the Visited BSM SHALL send the 'RoamingServiceRequest' (section 5.7.2.2) to the Home BSM. Home BSM SHALL respond to the request by sending 'RoamingServiceResponse' (section 5.7.2.3). In case the response allows roaming, then the Visited BSM sends a successful 'ServiceResponse' to the terminal. This leads to a subsequent LTKM delivery (push LTKM with Smartcard Profile or Trigger with DRM Profile).

The XML schema for these messages is defined in [BCAST10-XMLSchema-Roaming-backend].

5.7.2.1 Protocol stack for message exchanges between BSMs

The following protocol stack SHALL be used for message exchange between BSMs. HTTP over TCP/IP SHOULD be used for the delivery of the roaming procedure authorisation messages. HTTPS based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246] SHALL be used in conjunction with TCP/IP to provide secure delivery of the authorisation messages.



HTTP 1.1 over TCP/IP SHALL be used for file delivery via the interfaces, subject to the following conditions:

- The interfaces using HTTP 1.1 [RFC 2616] SHALL support gzip, compress, deflate and identity content codings. Other content codings MAY be supported.
- The interfaces using HTTP 1.1 [RFC 2616] MAY use persistent connections, pipelining and chunked transfer coding.

5.7.2.2 Back-end Interface Messages

Messages between the Visited and Home BSMs are transported using HTTP as the transport by placing both the requests and the responses addressed to either 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/bcast1.0/roaming HTTP/1.1\r\n<*requests*>

where the <host> denotes the part of the URI representing the address of the host.

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].

5.7.2.3 Processing and Responding

The processing of the messages between the Visited BSM and Home BSM involves first the HTTP transport level to deliver the messages between the Visited BSM and Home BSM. This is followed by the HTTP level passing the embedded XML message to the BSMs. While the status and error codes corresponding to the processing in the HTTP level are signaled using the HTTP headers, the result of the BSMs processing the XML request in the HTTP payload is signaled using XML messages placed into the payloads of the HTTP responses corresponding to the HTTP requests carrying the XML requests. Whenever an HTTP response contains an XML response from the BSM, the HTTP status code SHALL be set to 200 OK regardless of the contents of the XML response.RoamingServiceRequest

Name	Type	Category	Cardinality	Description	Data Type
RoamingServiceRequest	E			Request message for Roaming Service between Home BSM and Visited BSM. Contains the following attributes: requestID Contains the following elements: HomeBSMFilterCode VisitedBSMFilterCode TerminalSubscriptionType UserID GlobalPurchaseItemID	
requestID	A	M	1	An ID that is unique in the scope of this exchange that SHALL be used throughout the roaming service procedure. It SHALL be generated by the party that initiates the message exchange when it first requests roaming service.	unsignedInt
HomeBSMFilterCode	E1	M	1	The code that specifies the Home BSM. This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'BSMFilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSMFilterCode	E1	M	1	The code that specifies the Visited BSM. - This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'BSMFilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
Terminal Subscription Type	E1	M	1	A field that SHALL indicate the subscription scope of the terminal in terms of roaming. The Home Service Provider and the Visited Service Provider have a common understanding of the field according to roaming agreements between them. This element is not further specified in this specification.	anyURI
UserID	E1	M	1..N	A unique ID that SHALL be used to identify the terminal in both the Home Service Provider and Visited Service Provider BCAST service area. Contains the following attributes: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865]	unsignedByte

				1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	
GlobalPurchaseItemID	E1	M	1..N	Set of PurchaseItems (represented by GlobalPurchaseItemIDs) which are associated with the VisitedBSM and which the terminal wants to subscribe /to purchase.	anyURI

Table 40: Structure of RoamingServiceRequest Message

5.7.2.4 RoamingServiceResponse

Name	Type	Category	Cardinality	Description	Data Type
RoamingServiceResponse	E			Response message for Roaming Service between Home BSM and Visited BSM. Contains the following attribute: requestID roamingServiceStatus globalStatusCode Contains the following elements: UserID HomeBSMFilterCode VisitedBSMFilterCode GlobalPurchaseItemID	
requestID	A	M	1	An ID that is unique in the scope of this exchange SHALL be used throughout the roaming service procedure. It SHALL be generated by the party that initiates the message exchange when it first requests roaming service.	unsignedInt
roamingServiceStatus	A	M	1	A field that SHALL indicate whether the terminal has been authorized for roaming services or not. . The return codes are defined in section 5.11.	unsignedByte
globalStatusCode	A	M	0..1	The overall outcome of the request, according to the return codes defined in section 5.11. ▪ If this attribute is present and set to value “0”, the request was completed successfully. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘GlobalPurchaseItemID’. ▪ If this attribute is present and set to some other value than “0”, there was a generic error concerning the entire request. In this case the ‘itemwiseStatusCode’ SHALL NOT be given per each requested ‘GlobalPurchaseItemID’. If this attribute is not present, there was an error concerning one or more ‘GlobalPurchaseItemID’	unsignedByte

				elements associated with the request. Further, the 'itemwiseStatusCode' SHALL be given per each requested 'GlobalPurchaseItemID'.	
UserID	E1	M	1	A unique ID that SHALL be used to identify the terminal in both the Home Service Provider and Visited Service Provider BCAST service area. Contains the following attribute: type	string
type	A	M	1	Specifies the type of User ID. Allowed values are: 0 – username defined in [RFC 2865] 1 – IMSI 2 – URI 3 – IMPI 4 – MSISDN 5 – MIN 6-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
HomeBSMFilterCode	E1	M	1	The code that specifies the Home BSM. This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'BSMFilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
VisitedBSMFilterCode	E1	M	1	The code that specifies the Visited BSM. This element has the same structure as the 'BSMFilterCode' element in the 'ServiceGuideDeliveryDescriptor'.	complexType as defined for 'FilterCode' in section 5.4.1.5.2 of [BCAST10-SG]
GlobalPurchaseItemID	E1	M	0..N	Set of PurchaseItems (represented by GlobalPurchaseItemIDs) which are associated with the Visited BSM and which the terminal is authorized to subscribe /to purchase. This element SHALL NOT be instantiated in case the 'globalStatusCode' attribute is present and set to a value different from '0'. In any other case, it SHALL be instantiated. Contains the following attribute: itemwiseStatusCode	anyURI
itemwiseStatusCode	A	M	0..1	Specifies a status code of each GlobalPurchaseItemID using GlobalStatusCode defined in the section 5.11.	unsignedByte

Table 41: Structure of RoamingServiceResponse Message

5.7.3 Scope of identifiers and Home BSM identification while roaming

This section defines tools and normative requirements in order to address two potential scoping problems for identifiers while roaming:

1. When the terminal is in a roaming situation and issues service provisioning requests against the Home BSM, the latter may not be able to identify the services of the Visited BSM the user wishes to subscribe to. This is due to potential limitations in the scope of identifiers in the Service Guide, as such identifiers may not be globally defined. Said otherwise, the Home BSM may not be able to map fragment identifiers and global identifiers (such as the 'globalPurchaseItemId' attribute) back to the Visited BSM.
2. When the terminal is in a roaming situation and issues service provisioning requests against the Visited BSM, the latter may not be able to identify the Home BSM terminal despite the information provided by the terminal (such as device and user identifiers).

In order to address these potential problems, a 'BroadcastRoamingSpecificPart' element has been defined in the Service Provisioning messages, sections 5.1.5 and 5.1.6.

When in a roaming situation, the terminal issuing a Service Provisioning request against its Home BSM SHALL instantiate the 'VisitedBSM' element under the 'BroadcastRoamingSpecificPart' element for the said request.

When in a roaming situation, the terminal issuing a Service Provisioning request against the Visited BSM SHALL instantiate the 'HomeBSM' element under the 'BroadcastRoamingSpecificPart' element for the said request.

5.8 Availability of Location Information

BCAST Enabler MAY use Location Information for various purposes in conjunction with functions of BCAST, such as File and Stream Distribution and Service Guide. Location Information MAY be used to enable location based filtering of services; location based targeting of services; service blackout regions; and so on. It is out of scope of the BCAST Enabler how the Location Information is used by the functions of BCAST Enabler and what the exact behaviour of the terminal is. The following rules define the availability of Location Information to BCAST Enabler and the dependency of BCAST Enabler has with respect to Location Information:

- The BCAST system MAY utilize Location Information in OMA MLP format [OMA MLP].
- The BCAST system MAY utilize Location Information in BDS-specific cell_id (for example cell_id of 3GPP, 3GPP2, DVB-H, etc. system) format.
- The BCAST system MAY utilize Location Information in zip code format
- The BCAST system SHALL NOT expect all the BCAST terminals to have capability to utilize Location Information in either of the allowed formats.
- The method how BCAST terminal acquires the Location Information is out of scope of BCAST Enabler.
- The BCAST terminal MAY support the use of Location Information in OMA MLP format [OMA MLP].
- The BCAST terminal MAY support use the Location Information in BDS-specific cell_id format (for example cell_id of 3GPP, 3GPP2, DVB-H, etc. system)..
- The BCAST terminal MAY support use the Location Information in zip code format
- BCAST Service Guide MAY include the Location Information in the designated Service Guide fragments to specify the intended target area for BCAST Services. The Location Information MAY be included in either of the allowed formats, as define above. The exact specification on including the Location Information, refer to [BCAST10-SG]

The BCAST Terminal may have features or functionalities that are dependent on the availability of accurate location information. However, it is not in the scope of BCAST Enabler to ensure the availability of valid location information, Consequently, it is not in the scope of BCAST Enabler to enforce correct functioning of the features that are dependent on the location information.

5.9 XML for Signalling

The BCAST enabler uses XML as a format for many signalling messages (e.g. Service Guide Fragments, Provisioning Messages, Interactivity). This section describes how to facilitate a maximum degree of backward and forward compatibility between the current and future versions of BCAST. Furthermore, it ensures that vendor- and operator-specific extensions will not lead to inconsistent states when interpreting an XML instance. Related to this, design rules for extending XML schemas are given in Appendix G.

5.9.1 Namespace identifier

Each XML schema targets one XML namespace. The namespace identifiers of the BCAST XML schemas are structured as follows: <prefix>:<version>, where <prefix> is a colon-separated list of strings like “urn:oma:xml:bcast:sg:fragments” and <version> is the representation of the version of the BCAST enabler, structured as <major>.<minor>.<service_indicator>. While the <major> and <minor> parts of <version> SHALL be provided, the <service_indicator> part and its leading dot are OPTIONAL. A decoder SHOULD use <prefix> to determine that a particular piece of XML information is compliant with OMA BCAST, and SHOULD use <version> to determine its version.

5.9.2 Proprietary extensions

XML schemas defined in BCAST MAY be extended by proprietary elements. Such extensions SHALL be located inside a container called <PrivateExt> as defined in the XML schemas, and SHALL be defined in a non-BCAST namespace. Decoders MAY discard proprietary extensions. In any case, they SHALL NOT get into an error state when they encounter such extensions.

5.9.3 BCAST extensions

Decoders being able to interpret XML instances compliant to an earlier version of the OMA BCAST XML schemas but not able to interpret possible extensions MAY discard those extensions. In any case, they SHALL NOT get into an error state when they encounter unknown extensions.

5.10 Service Provisioning of Unicast Services

BCAST 1.0 enables a provider to offer services by both unicast and broadcast access methods. Service Provisioning for services that can be accessed via a Broadcast Channel typically involves Service and Content Protection [BCAST10-ServContProt]. Additionally, Service and Content Protection can be applied to services that can be accessed via the Interactive Channel. Alternatively, the access to those services can also be controlled by the BSM. In the latter case the BSM only allows access to the resource over the Interactive Channel after the user has purchased or subscribed to the associated purchase item of the service. So Service and Content Protection might not always be required for services that can be accessed via the Interactive Channel.

In such a case the terminal performs the regular Service Request and Service Response message sequence as defined in section 5.1.5.2. Upon successful purchase or subscription the ‘Service Response’ message from the BSM contains the ‘itemwiseStatusCode’ attribute set related to respective ‘PurchaseItemID’ set to ‘029’ (now subscribed). Further, in this case, the ‘DRMProfileSpecificPart’ element MAY be omitted. Upon reception of the request message the BSM MAY possibly proceed with the required charging event. Upon reception of the response message the terminal SHALL assume the network resource is accessible, i.e. the service can be consumed via the announced Access fragment in the Service Guide [BCAST10-SG].

5.11 Global Status Codes

The following table lists all the possible status codes for success or error case, and their applicability to each transaction. The table is to be used for GlobalStatusCode and roamingAuthorizationStatus in Provisioning and Roaming response messages. The codes may also be used in other response messages in other BCAST technical specifications.

Code	Status
000	**Success** The request was processed successfully.
001	**Device Authentication Failed** This code indicates that the BSM was unable to authenticate the device, which may be due to the fact that the device is not registered with the BSM, or that inappropriate security credentials were submitted by the device. In this case, the user may contact the BSM, and establish a contract, or get the credentials in place that are used for authentication.
002	**User Authentication Failed** This code indicates that the BSM was unable to authenticate the user, which may be due to the fact that the user is not registered with the BSM, or that inappropriate security credentials were submitted by the user. In this case, the user may contact the BSM, and establish a contract, or get the credentials in place that are used for authentication. Alternatively, if offered another opportunity, the user may re-enter the security credentials required for user authentication.
003	**Purchase Item Unknown** This code indicates that the requested purchase item is unknown. This can happen e.g. if the device has a cached service guide with old information. In this case, the user may re-acquire the service guide.
004	**Device Authorization Failed** This code indicates that the device is not authorized to get Long-Term Key Messages from the RI. For example, the device certificate was revoked in the case of the DRM Profile, or because trust relationship could not be established between the terminal and the BSM, in the case of the Smartcard Profile.
005	**User Authorization Failed** This code indicates that the user has not subscribed to the requested broadcast service, in the case of either the DRM Profile or the Smartcard Profile. In this case, the user may be given an opportunity to contact the BSM operator for service subscription.”.
006	**Device Not Registered** This code indicates that the device is not registered with the RI that is used for the transaction in the case of the DRM Profile, or that the device is not registered with the BDS-SD or the BSM, in the case of the Smartcard Profile. In this case, the device may automatically perform the registration, and, if the registration is successful, re-initiate the original transaction.
007	**Server Error** This code indicates that there was a server error, such as a problem connecting to a remote back-end system.
008	**Mal-formed Message Error** This code indicates that there has been a device malfunction, such as a mal-formed XML request. In such a case, the transaction may or may not (e.g. if there is an interoperability problem) succeed if it is re-initiated later. Note: This code can also be used between network entities
009	**Charging Error** This code indicates that the charging step failed (e.g. agreed credit limit reached, account blocked). The user may in such a case contact the BSM operator. Note: This code can also be used between network entities.

010	**No Subscription** This code indicates that there has never been a subscription for this service item, or that the subscription for this item has terminated. The user may in such a case issue a service request for a new subscription.
011	**Operation not Permitted** This code indicates that the operation that the device attempted to perform is not permitted under the contract between BSM and user. The user may in this case contact BSM operator and change the contract. Note: This code can also be used between network entities.
012	**Unsupported version** This code indicates that the version number specified in the request message is not supported by the network. In case the terminal cannot fall back to another version, the user may contact the BSM operator. Note: This code can also be used between network entities.
013	**Illegal Device** This code indicates that the device requesting services is not acceptable to the BSM. E.g. Blacklisted. In this case, the user may contact the BSM operator.
014	**Service Area not Allowed** This code indicates that the device is not allowed in the requested area due to subscription limits In this case, the user may contact the BSM operator or subscribe to the applicable service.
015	**Requested Service Unavailable** This code indicates that the requested service is unavailable due to transmission problems. In this case, the request may be re-initiated at a later time. Note: This code can also be used between network entities.
016	**Request already Processed** This code indicates that an identical request has been previously processed. In this case, the user or the entity may check to see if the request had already been processed (i.e. received an LTK), if not retry the request.
017	**Information Element Non-existent** This code indicates that the message includes information elements not recognized because the information element identifier is not defined or it is defined but not implemented by the entity receiving the message. In this case related entities should contact each other.
018	**Unspecified** This code indicates that an error has occurred which cannot be identified. In this case related entities should contact each other.
019	**Process Delayed** Due to heavy load, request is in the queue, waiting to be processed. In this case the user or entity should wait for the transaction to complete. Note: If this error occurs between network entities, the system should wait for the transaction to complete.
020	**Generation Failure** This code indicates that the request information (message) could not be generated. In this case the user or entity should retry later.

021	**Information Invalid** This code indicates that the information given is invalid and cannot be used by the system. In this case the request should be rechecked and sent again.
022	**Invalid Request** This code indicates that the requesting key materials and messages (e.g., LTKM) are not valid and can not be fulfilled. In this case the request should be rechecked and sent again.
023	**Wrong Destination** This code indicates that the destination of the message is not the intended one. In this case the request should be rechecked and sent again.
024	**Delivery of Wrong Key Information** This code indicates that the delivered key information and messages (e.g., LTKM) are invalid. In this case the request should be rechecked and sent again.
025	**Service Provider ID Unknown** This code indicates a confliction when the Visited or Home Service Provider requests a message to the Home or Visited Service Provider.
026	**Service Provider BSM_ID Unknown** This code indicates a confliction when the Visited or Home Service Provider BSM requests a message to the Home or Visited Service Provider BSM.
027	**Already in Use** This indicates requested setup value is already used in the Network Entity. Response message may contain the recommended value to use.
028	**No Matching Fragment** No fragment or SGDD matches the given request criteria.
029	**Now Subscribed** Specifies whether the subscription did succeed. Upon reception of this status code the terminal SHALL assume the service associated with the associated purchase item can be consumed via the associated 'Access' fragment of the service as defined in the Service Guide [BCAST10-SG]. This status code SHALL NOT be returned if the Purchase Item in question is associated with a service that is protected by Service or Content Protection.
030	**User already subscribed with different purchase options** Indicates that the user tries to repurchase an already subscribed item, but with different options. This can happen when terminal loses subscription information. In this case, the terminal MAY issue an AccountInquiry request to restore the subscription information.
031	**User must agree to the terms of use** Indicates that the BSM rejected the subscription because the user did not agree to the terms of use.
032 ~ 127	**Reserved for future use**
128 ~ 255	**Reserved for proprietary use**

Table 42: Global Status Codes

5.12 Auxiliary data download and insertion, and support for advertisements

The BCAST enabler supports the insertion of auxiliary data within the service in two ways. The first method is based on triggers that are delivered within notification messages. The second method is based on network operation for content delivery. Both methods are detailed below.

5.12.1 Auxiliary data insertion based on notification messages

This method is based on triggers that are delivered within notification messages. Such triggers can be used to trigger presentation of terminal-resident data or to initiate downloading of data to be presented later (in response to the trigger for auxiliary data insertion). These triggers are expected to produce terminal-specific behavior by specifying filtering data which may contain location context, target profiles, or which may reference terminal-resident rule sets. This results in selective downloading and insertion of auxiliary data, and can serve a variety of purposes, including personalization of the selection of terminal-resident advertisements for rendering. Triggers for auxiliary data downloading and insertion, and related signallingand message formats are normatively specified in chapter 5.14. The overall process normally consists of three steps.

1. An initial download notification trigger identifies an auxiliary data content item and an associated download opportunity.
2. At some future time, the terminal can download and store the corresponding auxiliary data content item (e.g. an advertisement) and store it locally. Such download SHALL be based on the delivery session information contained either in the initial trigger, or the Service Guide [BCAST10-ServiceGuide], and the decision whether or not to store the content is governed by filtering data contained in the download trigger.
3. Subsequently, an insertion notification trigger is sent, causing a suitable auxiliary content item (e.g. targeted advertisement) stored in the terminal to be played back during program viewing.

As indicated previously, the delivery session information for auxiliary data is conveyed in one of two ways:

a) For normal auxiliary data content, the Access/Session Description and Schedule fragments of the Service Guide MAY provide the associated delivery session information. In this mode, the File Delivery Client of the BCAST Terminal is responsible for downloading the associated file content.

b) Download session information MAY alternately be carried in the Notification message as described in Section 5.14. This represents a means to inform the terminal of dynamic updates to nominal auxiliary data content (e.g., delivery of last-minute changes to previously transmitted advertisements). In this mode, the Notification Client of the BCAST Terminal is responsible for downloading the associated file content.

The terminal determines the download method by examining the presence or absence of the element ‘SessionInformation’ in the Notification message. Presence of ‘SessionInformation’ indicates the use of delivery session information contained in the Notification message, whereas absence of this element indicates that the delivery session information is provided by the Service Guide.

For download session information specified in the Service Guide, the value of ‘ServiceType’ element in the Service Guide SHALL be “10” (i.e. correspond to “Auxiliary Data”). The existence of an ‘Auxiliary Data’ service SHOULD be hidden from user awareness in the Service Guide. The ‘Auxiliary Data’ service SHOULD be monitored by terminals which support auxiliary data download/caching for subsequent insertion display to users. Transport related information for downloading the associated auxiliary data files is provided either by the Access fragment that references the ‘Auxiliary Data’ service, or by the SessionDescription fragment identified by the Access fragment. Each content item comprising the Auxiliary Data service is scheduled for broadcast file delivery according to the Schedule fragment, and linked to the Notification message by the ‘GlobalContentID’ sub-element of the download ‘AuxDataTrigger’.

Once a notification carrying a ‘SessionInformation’ element is received, the associated auxiliary data download SHALL replace any auxiliary data download implied by session information in the Service Guide, or in earlier notification messages with a ‘SessionInformation’ element. Further updates to the same auxiliary data will therefore require further notifications with a ‘SessionInformation’ element.

5.12.2 Auxiliary data insertion based on network operation

This method is entirely based on network operation for content delivery. The network elements that schedule and transmit the service can perform the insertion of auxiliary data as normal content, multiplexed with the service. This method of auxiliary data insertion does not support rendering of terminal-resident auxiliary data nor does it allow personalization. The Service Guide data model inherently supports this method of auxiliary data insertion: auxiliary data can be added to an existing content or a new ‘Content’ fragment can be instantiated for auxiliary data.

5.13 Subtitling and Closed Captions

The Network MAY provide the subtitling or closed captions for a service using 3GPP Timed Text format. The Terminal SHOULD support 3GPP Timed Text as a format for subtitling and closed captions. The 3GPP Timed Text format is defined in [3GPP TS 26.245]. The signalling for subtitling is defined in section 5.1.2.5.2 of [BCAST10-SG].

5.14 Notification Function

Notification function can be used to provide information about forthcoming, imminent or immediate events, messages and notifications related to the BCAST system, to all broadcast services, or to a specific broadcast service. The notifications may be targeted to all reachable terminals or users, or specific terminals or users. Notifications are delivered as Notification Messages, which can be delivered over Broadcast Channel or over Interaction Channel, and stored in the terminal. Notification Messages fall into at least two categories, one category is user-oriented Notification Messages which are to be displayed to terminal users, the other category is terminal-oriented Notification Messages which are to be used for terminal operation and should not be displayed to users. The users are able to subscribe to user-oriented service-specific notifications using Service Provisioning Function specified in Section 5. Advertisement may be directly sent as Notification Messages, or triggered for local insertion by notification. The following outlines the purpose of Notification function in terms of types of Notification Messages that are specified:

- Emergency messages
- General announcements (informing about BCAST system problems, operator announcements, etc.)
- Broadcast main service or content associated notifications
  - Information regarding the availability of a specific service such as service breaks, abrupt change in the schedule (start time / end time) or access entry point of the service
  - Service-specific information that is a part of service experience (such as news, sports scores, etc.)
  - Information about services available in neighbouring systems, messages providing roaming support
  - Download or update announcement on SGDD or SG fragments
  - Download or update announcement on normal files such as movie, music, software, etc.
  - Auxiliary data downloading or insertion trigger (which are related to the main service or contents)
  - Other information related to the main service or content
- Notification-based information that the user has subscribed (i.e. asked to get delivered as soon the information is available).

Specification of Notification function consists of following parts:

- Discovery of availability and access to notifications
- Specification of event types of notifications (eventType)
- Format of Notification Message (syntax as defined by XML Schema in [BCAST10-XMLSchema-Notification])
- Notification Message delivery
  - Delivery over Broadcast Channel
  - Push delivery over Interaction Channel (including subscribing to notifications over Interaction Channel)
  - Polling notifications over Interaction Channel
- Notification interfaces(syntax as defined by XML Schema in [BCAST10-XMLSchema-Notification]).

Both the Network and Terminal MAY support the Notification function.

5.14.1 Discovery of Availability and Access to Notifications

5.14.1.1 Discovery of availability and access to general notifications

General notifications are not bound to any specific service nor Service Provider. Usually they are meant to be received by either all or majority of terminals, regardless of their Service Provider they are affiliated with. Examples of general notifications are emergency messages and announcements related to the operational aspects of BCAST system.

General notifications can be delivered either over Broadcast Channel or over Interaction Channel. The availability and access to general notifications can be discovered through SGDD.

5.14.1.1.1 General notifications: discovery through SGDD

The availability and access to general notifications can be signalled using the Service Guide Delivery Descriptor by instantiating the 'NotificationReception' element under the 'ServiceGuideDeliveryDescriptor' root element in the SGDD as defined in section 5.4.1.5.2 of [BCAST10-SG]. In case the Notification function is supported:

- NTC in the Terminal SHALL support the signalling of the availability and access to general notifications through the SGDD.
- NTDA in the Network SHALL support the signalling of the availability and access to general notifications through the SGDD.

5.14.1.2 Discovery of availability and access to notifications specific to a Service Provider

These notifications relate to a specific Service Provider. Usually they are meant to be received by either all or majority of terminals affiliated to the said Service Provider. Examples of such notifications are announcements related to operational aspects of a service.

Notifications specific to a Service Provider can be delivered either over Broadcast Channel or over Interaction Channel. The availability of and access to notifications specific to a Service Provider can be discovered through SGDD.

5.14.1.2.1 Notifications specific to a Service Provider: discovery through SGDD

The availability and access to notifications specific to a Service Provider can be signalled using the Service Guide Delivery Descriptor by instantiating the 'NotificationReception' element under the 'BSMSelector' element in the SGDD as defined in section 5.4.1.5.2 of [BCAST10-SG].

- NTC in the Terminal SHALL support the signalling of the availability and access to notifications specific to a Service Provider through the SGDD.
- NTDA in the Network MAY support the signalling of the availability and access to notifications a Service Provider through the SGDD.

5.14.1.3 Discovery of availability and access to service-specific notifications

Service-specific notifications are notifications that are associated with a specific service of a specific Service Provider. Usually they are meant to be received by the terminals that are accessing the service in question. Examples of service-specific notifications are sports goals, news and operational announcements related to a specific service.

Service-specific notifications can be delivered either over Broadcast Channel or over Interaction Channel. The availability and access to service-specific notifications can be discovered through 'Access' fragment.

5.14.1.3.1 Service-specific notifications: discovery through 'Access' fragment

The availability and access to service-specific notifications can be signalled by including the 'NotificationReception' element in any of the 'Access' fragments associated with a Service as defined in section 5.1.2.4 of [BCAST10-SG]. In case the Notification function is supported:

- NTC in the Terminal SHALL support the signalling of the availability and access to service-specific notifications through 'Access' fragment.
- NTDA in the Network SHALL support the signalling of the availability and access to service-specific notifications through the 'Access' fragment.

5.14.1.4 Discovery of availability and access to notifications as an independent service

Notification messages can also be delivered as part of a Notification service that can be discovered through the Service Guide. Usually they are meant to be received by the terminals as an independent service. Examples of such services are news tickers or traffic updates for navigation systems.

Independent Notification services can be delivered either over Broadcast Channel or over Interaction Channel. The availability and access to such Notification services can be discovered through the 'Service' and 'Access' fragments.

5.14.1.4.1 Notifications as an independent service: discovery through 'Access' fragment

The availability and access to independent Notification services is signalled by:

- providing a 'Service' fragment with the 'ServiceType' element set to '7' (Notification), and
- providing an 'Access' fragment pointing to the previous 'Service' fragment. The sessions or URLs used to deliver the Notification Messages SHALL be signalled by the "AccessType" element, by providing a session description of a file delivery session using the 'SessionDescription' element, or by providing a list of URLs to be polled for notification messages using the 'AccessServerURL' element. The 'NotificationReception' element MAY be instantiated (in order to declare the polling period over HTTP), but the IPBroadcastDelivery and PollURL SHALL not be given as this information is provided by the "AccessType" element.

In order to enable discovery of Notifications as independent services:

- NTC in the Terminal SHALL support the signalling of the availability and access to notification services through 'Service' and 'Access' fragment.
- NTDA in the Network SHALL support the signalling of the availability and access to notification services through the 'Service' and 'Access' fragment.

5.14.2 Declaring the usage of a Notification message

Besides the various elements and attributes of the Notification message schema, two attributes are used to specifically announce the intended usage of the message:

- the 'eventType' attibute describes the type of notification, it allows the Terminal to identify the nature of the received notification, and
- the 'notificationType' attribute, which is used to signal the primary target of the Notification message. As of this version 1.0 of the specification, either the notification message is primarily intended for the user (value '0') or the terminal (value '1'). A terminal-oriented notification usually implies preliminary processing of the notification message by the terminal prior to rendering, if any.

The table below defines the possible values of the 'eventType' attribute.

EventType	Name	Description
0	-	Reserved for future use.
1	Emergency notification	To announce emergency messages to users.
2	SG download or update notification	To announce download or update of SGDD or SG fragments
3	File download or update notification	To announce download or update of normal files such as movie, music, software, etc.
4	Service availability notification	To announce the errors, problems or interruption of broadcast main services or contents. To announce the abrupt schedule changes of broadcast main service or content

		To announce the abrupt changes on access entry point of broadcast main service or content.
5	Supplemental service notification	To announce service supplemental information that is a part of service experience (such as news, sports scores, promotional events etc.). Note that notifications of this type can either be related to a main service, or form a service on their own. In the latter case, i.e. when a service is of type "Notification", the type "Supplemental service notification" is the default "eventType" value of the Notification Messages in such a service. Note that a further version of this specification could specify new values to address additional use cases.
6	Auxiliary Data Trigger for Real-time main contents	To trigger either the auxiliary data downloading and storage, or the auxiliary data insertion, associated with the real-time main service or content. This notification may be associated with filtering related data to support customization of the auxiliary data storage or insertion.
7	Auxiliary Data Trigger for Non-Real-time main contents	To trigger either the auxiliary data downloading and storage, or the auxiliary data insertion, associated with the non-real-time main service content. This notification may be associated with filtering related data to support customization of the auxiliary data storage or insertion.
8 -127	Reserved for future use	
128 -255	Reserved for proprietary use	

Table 43: Event Types of Notifications

The various combination of ‘eventType’ and ‘notificationType’ are defined below. The reader is reminded that User-oriented notifications are signalled with value ‘0’ of the ‘notificationType’ attribute, while terminal-oriented notifications are signalled with value ‘1’.

Emergency notification (‘eventType’ value ‘1’)

User-oriented: the user is presented with the emergency notification. Other parameters of the notification message can impact the rendering of the notification, such as the ‘presentationType’ attribute.

Terminal-oriented: not applicable.

As of this version of the specification, the ‘notificationType’ attribute SHALL be set to ‘0’ if the ‘eventType’ attribute is set to ‘1’. In case the ‘notificationType’ attribute is not set to ‘0’ but the eventType attribute is set to ‘1’, the 1.0 terminal SHALL assume the notification message is a User-oriented emergency message and SHALL proceed as per the rules defined in section 5.14.6.

Service availability notification (‘eventType’ value ‘4’)

User-oriented: the user is presented with the payload of the notification.

Terminal-oriented: not applicable.

As of this version of the specification, the ‘notificationType’ attribute SHALL be set to ‘0’ if the ‘eventType’ attribute is set to ‘4’. In case the ‘notificationType’ attribute is not set to ‘0’ but the eventType attribute is set to ‘4’, the 1.0 terminal MAY discard the message.

Supplemental service notification (‘eventType’ value ‘5’)

User-oriented: the user is presented with the payload of the notification.

Terminal-oriented: not applicable.

As of this version of the specification, the ‘notificationType’ attribute SHALL be set to ‘0’ if the ‘eventType’ attribute is set to ‘5’. In case the ‘notificationType’ attribute is not set to ‘0’ but the eventType attribute is set to ‘5’, the 1.0 terminal MAY discard the message.

Auxiliary Data Trigger for Real-time main content (‘eventType’ value ‘6’)

User-oriented: not applicable.

Terminal-oriented: the terminal silently processes the payload of the notification.

As of this version of the specification, the ‘notificationType’ attribute SHALL be set to ‘1’ if the ‘eventType’ attribute is set to ‘7’. In case the ‘notificationType’ attribute is not set to ‘1’ but the eventType attribute is set to ‘7’, the 1.0 terminal MAY discard the message.

Auxiliary Data Trigger for Non-Real-Time man content (‘eventType’ value ‘7’)

User-oriented: not applicable.

Terminal-oriented: the terminal silently processes the payload of the notification.

As of this version of the specification, the ‘notificationType’ attribute SHALL be set to ‘1’ if the ‘eventType’ attribute is set to ‘8’. In case the ‘notificationType’ attribute is not set to ‘1’ but the eventType attribute is set to ‘8’, the 1.0 terminal MAY discard the message.

5.14.3 Format of Notification Message

Notification Message structure consists of:

- **Generic fields:** id, version, notificationType, eventType, IDRef, validTo, Title, Description, PresentationType and Extension
- **Notification content:** SessionInformation, MediaInformation, SGDD, SGDDReference, FragmentReference and AuxDataTrigger

While the generic fields can be used with all types of notifications, the notification content varies according to the notification type and event type. For example: emergency notification could contain generic fields + MediaInformation; SG download or update notification could contain SGDD, SGDDReference, or FragmentReference, etc.

A Notification Message carrying Service Guide update (eventType with value 2) SHALL only notify updates that relate to the currently bootstrapped Service Guide.

Name	Type	Category	Cardinality	Description	Data Type
Notification Message	E			Notification Message Contains the following attributes: id version notificationType eventType validTo Contains the following elements:	

				IDRef Title Description PresentationType Extension SessionInformation MediaInformation ServiceGuide AuxDataTrigger PrivateExt	
id	A	NM/ TM	1	Identifier of Notification Message	anyURI
version	A	NM/ TM	1	Notification Message version information. It is to be used to check for Notification Message Redundancy and new Notification Messages. This field can be expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
notificationType	A	NM/ TM	1	This element indicates whether the message is primarily targeted at the user or the terminal. Allowed values are: 0 – indicates that this message is user-oriented 1– indicates this message is terminal-oriented 2 - 127: For future use 128 - 255: For proprietary use Possible combinations of the ‘notificationType’ attribute and the ‘eventType’ attribute are detailled in section 5.14.2.	unsignedByte
eventType	A	NM/TM	1	This element indicates the nature of the notification conveyed by the message. For a detailed list of values of the ‘eventType’ attribute and its possible combinations with ‘notificationType’, see section 5.14.2.	unsignedByte
validTo	A	NM/ TM	0..1	Valid time of Notification Message. This field expressed as the first 32bits integer part of NTP time stamps. If ‘validTo’ is specified, the Notification Message SHOULD be expired at the specified time.	unsignedInt
IDRef	E1	NM/ TM	0..N	Fragment ID references of the main services or contents which the Notification Message is related to. This SHALL only be used for Service specific or AuxData Notifications. The terminal SHALL consider the fragment reference to be scoped under the BSMSelector the Notification message applies to.	anyURI
Title	E1	NM/ TM	0..N	Title of Notification Message, 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 or Messages of Notification, possibly in multiple languages The language is expressed using built-in XML attribute ‘xml:lang’ with this element	string

				Only one instance of this element is allowed per language.	
Presentation Type	E1	NM/ TM	0..1	Recommends the type of presentation for the received Notification Messages based on the priority of the Notification Message. Allowed values are: 0 – For high priority Notification Messages, Terminal MAY immediately render the message after interrupting all the applications. 1 – For medium priority Notification Messages, Terminal MAY immediately render the message, overlaying the present playing services. 2 – For low priority Notification Messages, Terminal MAY NOT immediately render the message, the user can see the stored message whenever he or she wants. 3-127: For future use 128-255: For proprietary use	unsignedByte
Extension	E1	NM/ TM	0..N	Additional information related to this Notification Message. Contains following attribute: url Contains following sub-element: Description	
url	A	NM/ TM	1	URL containing additional information related to this notification.	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
SessionInformation	E1	NM/ TM	0..1	This element SHALL be present when the Notification Message carries pointer to another delivery session, for example for file download or update, SG download or update, or auxiliary data download. SessionInformation defines the delivery session information, including the schedule, of the objects delivered over the broadcast channel, and URI as alternative method for delivery over interaction channel. After receiving Notification Message with SessionInformation, Terminal would access the relevant session specified by SessionInformation and take a proper action like receiving contents. Contains the following attributes: validFrom validTo usageType Contains the following elements:	

				DeliverySession AlternativeURI Nominally, access and schedule related delivery session information for relatively long-lived auxiliary data contents SHOULD be specified by the Access and Schedule fragments, respectively, of the Service Guide [BCAST10-ServiceGuide]. Other updates of auxiliary data MAY be delivered on the delivery session referenced by this SessionInformation, specifically, over the duration spanned by the ('validFrom', 'validTo') attributes of this element. If 'AuxDataTrigger' is instantiated and corresponds to the download trigger, and 'SessionInformation' is also instantiated, then the latter element SHALL be used to convey the delivery session information for the auxiliary data content. If 'AuxDataTrigger' is instantiated and corresponds to the download trigger, and 'SessionInformation' is not instantiated, then the <GlobalContentID> sub-element of 'AuxDataTrigger' SHALL be instantiated to provide the linkage to the Service Guide, which in turn conveys the delivery session information for the auxiliary data content. Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first SessionInformation element.	
validFrom	A	NM/ TM	0..1	The first moment when the session for terminal to receive data is valid. This field expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
validTo	A	NM/ TM	0..1	The last moment when the session for terminal to receive data is valid. This field expressed as the first 32bits integer part of NTP time stamps.	unsignedInt
usageType	A	NM/ TM	0..1	Defines the type of the object transmitted through the indicated delivery session. Allowed values are: 0 – unspecified 1 - files 2- streams 3 – SGDD only 4 – mixed SGDD and SGDU 5 - notification 6-127 reserved for future use 128-255 reserved for proprietary use Note: the delivery session only carrying SGDUs is declared through 'SGDD' element or	unsignedByte

				"SGDDReference" element in this Notification Message. Default: 0	
Delivery Session	E2	NM/ TM	0..1	Target delivery session information indicated by the Notification Message. Contains the following attributes: ipAddress port sourceIP transmissionSessionID Contains the following element: 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
sourceIP	A	NM/ TM	0..1	Source IP address of the delivery session	string
transmission SessionID	A	NM/ TM	1	This is the Transmission Session Identifier (TSI) of the session at ALC/LCT level.	unsignedShort
ContentLocation	E3	NM/TM	0..1	This is the location of the Content to be retrieved. It corresponds to the 'Content-Location' attribute in the FDT. Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first ContentLocation element.	anyURI
AlternativeURI	E2	NM/ TM	0..1	Alternative URI for receiving the object via the interaction channel. If terminal cannot access the indicated delivery session, the terminal can receive the objects associated with the Notification Message by AlternativeURI. Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first AlternativeURI element.	anyURI
Media Information	E1	NO/ TM	0..1	This element SHALL be present when the Notification Message carries information for rendering support of the notification. Media Information is used to construct and render Notification Messages. The notification media objects declared below can be delivered over a file delivery session specified	

				by 'DeliverySession' element, or be retrieved via interaction channel via AlternativeURI of the media object as defined below. Contains the following elements: DeliverySession Picture Video Audio	
DeliverySession	E2	NM/ TM	0..1	Session information to retrieve contents to be used for MediaInformation. Contains the following elements: ipAddress port sourceIP transmissionSessionID	
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
sourceIP	A	NM/ TM	0..1	Source IP address of the delivery session	string
transmission SessionID	A	NM/ TM	1	This is the Transmission Session Identifier (TSI) of the session at ALC/LCT level.	unsignedShort
Picture	E2	NO/ TM	0..1	Defines how to obtain a picture and MIME type. Contains the following attributes: mimeType pictureURI Contains the following element: AlternativeURI Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first Picture element.	
mimeType	A	NO/ TM	0..1	MIME type of Picture	string
pictureURI	A	NO/ TM	0..1	This is the location of the Content to be retrieved. It corresponds to the 'Content-Location' attribute in the FDT, if FLUTE is used to deliver the pictureURI. This attribute is to be used in conjunction with the DeliverySession element defined above.	anyURI
AlternativeURI	E3	NO/ TM	0..N	Alternative URI for receiving the object via the interaction channel. If terminal cannot access the indicated delivery session, the terminal can receive the objects associated with the Notification Message by AlternativeURI. Multiple instances of the AlternativeURI MAY be instantiated for the purpose of server load	anyURI

				distribution.	
Video	E2	NO/ TO	0..1	Defines how to obtain a video and MIME type. Contains the following attributes: mimeType codec videoURI Contains the following element: AlternativeURI Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first Video element.	
mimeType	A	NO/ TO	0..1	MIME type of Video	string
codec	A	NO/ TO	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 [RFC 4281].	string
videoURI	A	NO/ TO	0..1	This is the location of the Content to be retrieved. It corresponds to the 'Content-Location' attribute in the FDT, if FLUTE is used to deliver the videoURI. This attribute is to be used in conjunction with the DeliverySession element defined above.	anyURI
AlternativeURI	E3	NO/ TM	0..N	Alternative URI for receiving the object via the interaction channel. If terminal cannot access the indicated delivery session, the terminal can receive the objects associated with the Notification Message by AlternativeURI. Multiple instances of the AlternativeURI MAY be instantiated for the purpose of server load distribution.	anyURI
Audio	E2	NO/ TM	0..1	Defines how to obtain a audio and MIME type. Contains the following attributes: mimeType codec AudioURI Contains the following element: AlternativeURI	

				Note: For BCAST 1.0 the cardinality is set to 0..1 however in future releases the cardinality may change to 0..N in this case for backward compatibility the 1.0 terminal SHOULD only utilize the first audio element.	
mimeType	A	NO/ TM	0..1	MIME type of Audio	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 audio/3GPP, audio/3GPP2 is specified in [RFC 4281].	string
audioURI	A	NO/ TM	0..1	This is the location of the Content to be retrieved. It corresponds to the 'Content-Location' attribute in the FDT, if FLUTE is used to deliver the audioURI. This attribute is to be used in conjunction with the DeliverySession element defined above.	anyURI
AlternativeURI	E3	NO/ TM	0..N	Alternative URI for receiving the object via the interaction channel. If terminal cannot access the indicated delivery session, the terminal can receive the objects associated with the Notification Message by AlternativeURI. Multiple instances of the AlternativeURI MAY be instantiated for the purpose of server load distribution.	anyURI
ServiceGuide	E1	NO/ TO	0..N	This element acts as a placeholder for future extensions of the Notification message in order to allow notifications related to the Service Guide (e.g. Service Guide update). BCAST 1.0 Terminals MAY decide to perform a complete Service Guide update in case they receive a Notification message with value '2' of the 'eventType' attribute and an instance of the 'ServiceGuide' element.	ServiceGuideType as defined in section 5.14.8
AuxDataTrigger	E1	NO/ TO	0..N	This Element contains information to trigger the auxiliary data downloading and storage, or the auxiliary data insertion associated with main service or content. 'globalContentID' and/or 'FilteringData' can be used to identify and/or fetch the auxiliary data content, and/or FilteringData associated with the auxiliary data content. Note: The auxiliary data downloading trigger indicates that auxiliary data should be downloaded and stored when the filtering criteria are met. Absence of FilteringData in the downloading trigger implies that the auxiliary	

				data should be stored. Persistence of storage is terminal implementation dependent. Contains the following attributes: type Contains the following Elements: GlobalContentID FilteringData PresentationRule	
type	A	NM/TM	1	0-The auxiliary data trigger is a download trigger; 1- The auxiliary data trigger is an insertion trigger. 2-127: Reserved for future use. 128-255: Reserved for proprietary use When AuxDataTrigger is present in the Notification Message, and the ‘type’ attribute is set to ‘0’, the IDRef element SHALL NOT be present; When the ‘type’ attribute is set to ‘1’, the IDRef element SHALL be present to indicate to terminal which main service should this auxiliary data be inserted to.	unsignedByte
GlobalContentID	E2	NO/ TM	1	Globally Unique Identifier of the auxiliary data content. If ‘AuxDataTrigger’ is of <type> = 0 (download trigger), and o if ‘SessionInformation’ is absent, then this element's value SHALL match that of the <globalContentID> attribute of a Service Guide Content fragment which describes a content item belonging to the ‘Auxiliary Data’ service type (see [BCAST10-ServiceGuide]). ‘GlobalContentID’ then identifies the auxiliary data content item to which the ‘FilteringData’ sub-element of ‘AuxDataTrigger’ is applicable in controlling the subsequent download and caching operation. o if 'SessionInformation' is present, then ‘GlobalContentID’ is used to identify the Auxiliary Data content downloadable via the 'SessionInformation', for the purposes of replacing Auxiliar Data content already on the terminal, or cancelling a later scheduled download of that Auxiliary Data content using the Service Guide. ‘GlobalContentID’ is not intended to be used to point at the service guide for providing download session information. If ‘AuxDataTrigger’ is of <type> = 1 (insertion trigger), then this element SHALL be omitted. Such insertion trigger and its associated filtering data leads to the insertion of terminal-resident	anyURI

				auxiliary data for rendering (which is independent of the identification of an auxiliary data content item for downloading and caching purposes).	
FilteringData	E2	NO/ TO	0..N	Reference to the location of the filtering related information associated with the AuxDataTrigger Notification Message, or the filtering–related information embedded within this Notification Message. Note: filtering related information can include attributes, values, rules, filter IDs, etc. Contains the following sub-elements: Location TargetProfile FilterIDs Either Location, TargetProfile, or FilterIDs, but not more than one of these sub-elements, MAY be present in FilteringData.	
Location	E3	NO/ TM	0..1	Reference to the location of the filtering related information associated with the AuxDataTrigger, from which that data can be retrieved.	anyURI
TargetProfile	E3	NO /TM	0..N	Filter rules and/or attributes to be used in the selection of auxiliary data for downloading and storage, or insertion. The extensible list of TargetProfile for a particular AuxDataTrigger notification enables the filtering/customization of the auxiliary data triggered by the notification, according to any specified filtering characteristic, e.g. user preference, user age, user location, service provider, etc. If the AuxDataTrigger is used to trigger the terminal to download and cache auxiliary data, in which case the ‘type’ attribute under AuxDataTrigger is set to ‘0’, the number of TargetProfile entries SHALL be the same as the number of SessionInformation entries, and specifically, TargetProfile 1 maps to SessionInformation 1, TargetProfile 2 maps to SessionInformation 2, and so on. Attribute: filterID Sub-elements: Attribute FilterRules Note: TargetProfile is intended to be used to identify the type of auxiliary data file associated with the AuxDataTrigger notification. As an example, for an ad insertion event, ‘attributeName’ = “URI” and ‘attributeValue’ = “advertisement” can be used to match against the URI identifiers of auxiliary data files stored on the terminal for the keyword “advertisement”. Such mechanism would identify all the advertisements stored on the terminal, for	

				subsequent insertion selection based on filter rules/attributes.	
filterID	A	NO /TM	0..1	Identity of the TargetProfile to be stored on the terminal for subsequent reference as a Filter ID sent as part of the FilterIDs (E3).	anyURI
Attribute	E4	NO/ TM	0..N	Profile attribute. Contains the following attributes: name value	
name	A	NM/TM	1	Profile attribute name	string
value	A	NM/ TM	1	Profile attribute value.	string
FilterRules	E4	NM/ TM	0..1	Filter rules that are used in the selection of auxiliary data for downloading and storage, or insertion.	string
FilterID	E3	NO /TM	0..N	Zero or more filter IDs used in the selection of auxiliary data for downloading and storage, or insertion. Each ad filter ID is an alias for a corresponding set of filter rules stored in the terminal. The rule set(s) in the FilterID list is(are) applied to the selection of the auxiliary data for downloading and storage, or insertion. The FilterID refers to the TargetProfile previously stored on the terminal.	anyURI
Presentation Rule	E2	NO/ TM	0..1	Specifies the presentation rules when the cached content should be rendered with this Notification Message. Contains the following attributes: renderingTime duration	
renderingTime	A	NO/ TM	0..1	Specifies the timing to start the presentation of the auxiliary data. In case eventType = 6 this element represent the time instant as the first 32bits integer part of NTP time for which the Notification Message is displayed or the auxiliary data insertion event occurs. In case eventType = 7, this element represent the offset in segments for which the auxiliary data insertion event occurs, relative to the start of the presentation of the associated main content.	unsignedInt
duration	A	NO/ TM	0..1	Time length of presentation of the auxiliary data in seconds.	unsignedShort
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.	

Table 44: Structure of Notification Message

5.14.4 Notification Message Delivery

Notification Messages are created by the NTG (Notification Generation Function) according to the structure in 7.3 and are prepared for delivery by the NTDA (Notification Distribution/Adaptation Function). Notification Messages MAY be delivered in a number of ways:

- Notification Message delivery over Broadcast Channel (see section 5.14.4.1)
- Notification Message push-delivery over Interaction Channel (see section 5.14.4.2)
    - Related to push-delivery over Interaction, subscribing to Notification Messages (see section 5.14.4.2.1)
- Polling Notification Messages over Interaction Channel (see section 5.14.4.3)

5.14.4.1 Notification Message Delivery over Broadcast Channel

Over Broadcast Channel, the Notification Messages SHALL be delivered to terminals using one of the following methods:

1) UDP delivery: The Notification Message is delivered in a UDP packet.

The UDP packet SHALL be sent over the Broadcast Channel using the UDP destination port defined in the NotificationReception in the SGDD or the ‘Access’ fragment and the IP address of the ongoing session that the Notification Message is targeted for. If a separate IP address is defined in the NotificationReception in the SGDD or ‘Access’ fragment for the Notification Message then it SHALL be used. It is RECOMMENDED that to avoid IP level segmentation, Notification Message sizes should be less than 1500 bytes, the average network MTU (Maximum Transfer Unit) size.

To decrease the message size, GZIP MAY be used to compress the Notification Message.

The payload of the UDP file SHALL start with a header as specified below, followed by the uncompressed or compressed Notification Message. The format of the header is defined as follows:

Field	Type	Definition
Payload_type	uimsbf4	Signals the type of the payload Values: 0 – Notification according to MIME type vnd.oma.bcast.notification+xml 1-7 – reserved for future BCAST extensions 8-15 – reserved for proprietary extensions
Encoding_type	uimsbf4	Signals the encoding of the payload Values: 0 – unencoded 1 – GZIP encoded 2-7 – reserved for future BCAST extensions 8-15 – reserved for proprietary extensions

Table 45: Header for UDP Delivery of Notification Message

Mnemonics: uimsbf4 = Unsigned 4 bit Integer, most significant bit first

2) File delivery: The Notification Message is delivered in a separate file delivery session. There are two options for announcing such a file delivery session.

(a) the session parameters are announced in a separate Notification Message using the ’DeliverySession’ element. This approach is RECOMMENDED for service-specific notifications, general notifications, and notifications specific to a Service Provider in case the Notification Message size exceeds the MTU size.

(b) the session parameters are announced in the Service Guide. This method SHALL be used for independent Notification services as specified in section 5.14.1.4.

To decrease the message size, GZIP MAY be used to compress the Notification Message. The fact that a message is compressed SHALL be signalled in the FDT. The Content-Type of a Notification Message in the FDT SHALL be signalled as “application/vnd.oma.bcast.notification+xml".

The terminal SHALL support GZIP decompression of Notification Messages.

The Notification Messages MAY be repeatedly transmitted by the Service Provider or Network Provider to increase the probability of all intended terminals receive the Notification Messages.

The following figures illustrate the protocol stacks of the two Notification Message delivery methods over the Broadcast Channel:

Notification Message

ALC/LCT/FLUTE

UDP

IP

Figure 1: Notification message delivery protocol stack variant 1

Notification Message

UDP

IP

Figure 2: Notification message delivery protocol stack variant 2

5.14.4.2 Notification Message push-delivery over Interaction Channel

The NDTA MAY deliver a Notification Message to the NTC using OMA Push as defined in [BCAST10-Distribution]. The terminal MAY support reception of Notification Messages delivered with OMA Push as defined in [BCAST10-Distribution].

5.14.4.2.1 Subscribing and Unsubscribing to User-oriented Notification Messages

Service Provisioning Function SHOULD be used for subscribing or unsubscribing Notification Message over Interaction channel. If the terminal has interaction capability, the terminal SHOULD support subscription and unsubscription of Notification Messages.

- When Terminal subscribes service-specific notification or notification service, Service Request message (See section 5.1.5) SHALL include ‘ServiceID’ element and ‘notification’ attribute under ‘ServiceID’ element
- When Terminal unsubscribes service-specific notification or notification service, Unsubscription message (See section 5.1.5) SHALL include ‘keepSubscription’ attribute, ‘ServiceID’ element and ‘notification’ attribute under ‘ServiceID’ element.

5.14.4.3 Polling notifications over Interaction Channel

In case the Terminal supports the Notification function, the NTC in Terminal with Interaction Channel capability SHALL support polling to notifications over Interaction Channel as follows:

- NTC sends an HTTP1.1 GET Request to the NTDA that is signalled in SGDD or ‘Access’ fragment.
- Response to the HTTP Request sent to the NTDA SHALL embed a set of zero or one or more Notification Messages encapsulated in a Notification MessageContainer delivered over an HTTP1.1 message. The NotificationMessageContainer embeds the exhaustive list of valid Notification Messages (i.e. that have not

expired at the time of the response). The following table specifies the format of the NotificationMessageContainer.

Name	Type	Category	Cardinality	Description	Data Type
Notification MessageContainer	E			Notification MessageContainer Contains the following element: NotificationMessage	
Notification Message	E1	NM/ TM	0..N	Notification Message as specified in section 5.14.3	Notification MessageType as specified in section 5.14.3

- The Content-Type of the HTTP Response message containing a NotificationMessageContainer SHALL be set to “application/vnd.oma.bcast.notification+xml”.
- In order to reduce the information that are present in the NTDA response when no update occurred between two subsequent requests from the same terminal, it is recommended that the NTDA sends an entity tag and/or a Last-Modified value in each HTTP 1.1 [RFC 2616] response header. It is expected that the NTC uses these information for subsequent cache-conditional requests as specified in HTTP 1.1 [RFC 2616].
- As specified in HTTP 1.1 [RFC 2616], if the NTC has performed a conditional GET request and access is allowed, but the document has not been modified, the NTDA SHOULD respond with the 304 status code. The 304 response SHALL NOT contain a message-body, and thus is always terminated by the first empty line after the header fields.
- As specified in [BCAST10-SG], when the "PollPeriod" element is instantiated in the "NotificationReception" element of an Access fragment or an SGDD, no caching mechanisms of HTTP 1.1 [RFC 2616] SHOULD conflict with the fact that the NTC is expected to request for a fresh NotificationMessageContainer every "PollPeriod" value.
- The NTDA 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”. Terminals SHALL support this content encoding.

5.14.5 Notification Interfaces

The following sections specify the Notification interfaces between logical BCAST “backend” entities for message exchanges. 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 can be realized using protocols and methods not specified here. If a BCAST implementation does expose the interfaces, the network SHALL support the Notification Backend Interfaces syntax as defined by XML Schema in [BCAST10-XMLSchema-Notification].

5.14.5.1 Protocol Stacks

The following protocol stack SHALL be used for exchanging messages between Notification Components such as CC, NTE, NTG, and NTDA. HTTP or HTTPS that SHALL be based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246] over TCP/IP SHALL be used for the delivery of messages.

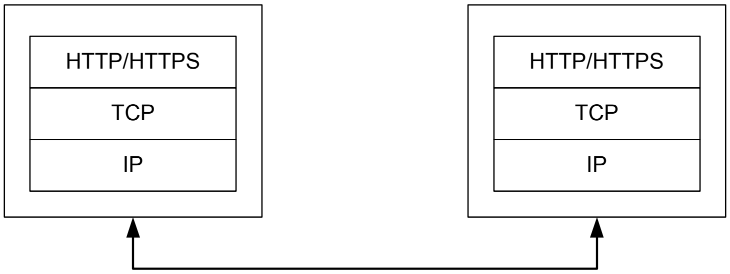


Figure 3: Noticifcation component exchange protocol stack

Messages to and from CC, NTE, NTG or NTDA are transported using HTTP by placing both the requests and the responses addressed to CC, NTE, NTG or NTDA 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/bcast1.0/nt HTTP/1.1\r\n<NTEReq>
- POST <host>/oma/bcast1.0/nt HTTP/1.1\r\n<NTDReq>

where the <host> denotes the part of the URI representing the address of the host.

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].

5.14.5.2 Notification Event Delivery

Notification Event can be generated in CC, BSA, BSM, or BSD/A. Each Entity delivers Notification Event via Backend Interface such as NT-1, NT-3, and NT-4. CC can deliver Notification Event to NTE via NT-1, NTE will deliver Notification Event generated in either CC or BSA to NTG via NT-3, and NTDA will deliver Notification Event generated in BSD/A to NTG via NT-4.

5.14.5.2.1 Request Message

The following is the delivery message of Notification Event, which is sent from the CC (Content Creation) to the NTE over interface NT-1, from NTE to NTG over interface NT-3 or NTDA to NTG over interface NT-4.

Name	Type	Category	Cardinality	Description	Data Type
NTEReq	E			Specifies the delivery message of Notification Event for generating Notification Message. Contains the following attributes: nteID entityAddress deliveryPriority Contains the following elements: NotificationEvent	
nteID	A	M	1	Identifier of Notification Event	unsignedInt
entityAddress	A	M	1	Network Entity Address to receive the response of this message.	anyURI
deliveryPriority	A	O	0..1	Defines the priority of this notification event. This information is applied to generate Notification Message in NTG. NTG may be ignored this field.	boolean
NotificationEvent	E1	M	1..N	Specifies the Notification Event, containing information to be included into the Notification Message. It is RECOMMENDED that the information is delivered in the form of BCAST Notification Message format (as specified in section 5.14.3). Other formats MAY be used only for NT-1.	

				Contains the following sub-element: NotificationMessage	
Notification Message	E2	O	0..1	BCAST NotificationMessage format as specified in section 5.14.3. The following rule applies to child elements or attributes of NotificationMessage which are not relevant: If the element/attribute has a minimum cardinality of 0, it SHALL NOT be instantiated. Otherwise, it SHALL be delivered as empty field.	complexType as specified in section 5.14.3
Private	E2	O	0..1	This container allows to use data formats not specified in BCAST.	

Table 46: Structure of Notification Event Request Message

5.14.5.2.2 Response Message

The following is the response message of NotificationEvent Delivery and which is sent from the NTE to CC over interface NT-1, from NTG to NTE over interface NT-3 or form NTG to NTDA over interface NT-4.

Name	Type	Category	Cardinality	Description	Data Type
NTERes	E			Specifies the Response message for NTEReq. Contains the following elements: Result	
Result	E1	M	1..N	The list of results, each entry consisting of a pair of ID and statusCode Contains the following attributes: nteID statusCode	
nteID	A	M	1	Identifier of NTEReq Message	unsignedInt
statusCode	A	M	1	Indicates the overall outcome how NTEReq is processed, according to the global status code (as specified in Section 5.11).	unsignedByte

Table 47: Structure of Notification Event Response Message

5.14.5.3 Notification Message Delivery

Notification Message is generated by NTG in BSM. NTG will request to deliver Notification Message to NTDA via NT-4.

5.14.5.3.1 Request Message

The following is the delivery message of Notification Message which is sent from the NTG to NTDA over interface NT-4.

Name	Type	Category	Cardinality	Description	Data Type
NTDReq	E			Specifies the Request message of Notification Message Delivery from NTG to NTDA. Contains the following attributes: ntdReqID entityAddress deliveryPriority Contains the following elements: TargetAddress NotificationMessage	

ntdReqID	A	M	1	Identifier of NTDReq	unsignedInt
entityAddress	A	M	1	Network Entity Address to receive the response of this message.	anyURI
deliveryPriority	A	O	0..1	Defines the delivery priority of this Notification Message. NTG can request NTDA to deliver this notifcaiton message as high priority. If priority=true, it means high priority. If priority=false, it means general message.	boolean
TargetAddress	E1	O	0..N	Specifies TargetAddress to deliver Notification Message. For service-specific notification, AccessReference or address under NotificationReception in 'Access' fragment can be possible value. If Notification message is delivered over interaction channel, the value can be e-mail address, IMSI, etc. If not given, Notification message SHALL be delivered to all users of the service provider using address defined in SGDD. Contains the following attributes: deliveryChannel AddressType	string
deliveryChannel	A	M	1	Specifies the delivery channel If deliveryChannel = false, Notificaiton Message SHALL be delivered over Broadcast Channel. If deliveryChannel = true, Notification Message SHALL be delivered over Interaction Channel.	boolean
addressType	A	M	1	Specifies the type of TargetAddress Value 0 - IPAddress 1 - anyURI 2 - IMSI 3 -127: For Future Use 128 - 255: For Proprietary Use	unsignedByte
Notification Message	E1	O	0..1	BCAST NotificationMessage format as specified in section 5.14.3. The following rule applies to child elements or attributes of NotificationMessage which are not relevant: If the element/attribute has a minimum cardinality of 0, it SHALL NOT be instantiated. Otherwise, it SHALL be delivered as empty field.	complexType as specified in section 5.14.3

Table 48: Structure of Notification Delivery Request Message

5.14.5.3.2 Response Message

The following is the response message of Notification Message Delivery which is sent from NTDA to NTG over interface NT-4.

Name	Type	Category	Cardinality	Description	Data Type
NTDRes				Specifies the Response message for NTDReq. Contains the following elements:	

				Result	
Result	E1	M	1..N	The list of results, each entry consisting of a pair of request ID and statusCode Contains the following attributes: ntdReqID statusCode	
ntdReqID	A	M	1	Identifier of NTDReq Message	unsignedInt
statusCode	A	M	1	Indicates the overall outcome how NTDReq is processed, according to the global status code (as specified in Section 5.11).	unsignedByte

Table 49: Structure of Notification Delivery Response Message

5.14.6 Minimal support for emergency notifications

If the terminal supports emergency notifications, then the terminal SHALL support the use of Notification Function for those notifications as follows:

- The terminal SHALL be able to discover of the entry point to notification delivery channel as specified in section 5.14.1.1.1 through the use of element 'NotificationReception'. Further, the terminal SHALL assume that 'NotificationReception' element describes the entry point to general notification delivery channel within which the notification messages are delivered using "UDP Delivery" as specified in section 5.14.4.1.
- The terminal SHALL support the "UDP delivery" over Broadcast Channel as specified in section 5.14.4.1 as follows:
  - The terminal SHALL support 'Payload_type' having value '0'
  - The terminal SHALL support 'Encoding_type' having value '0'
- The terminal SHALL support the Notification Message format for emergency notifications as follows:
  - The terminal SHALL assume that attribute 'notificationType' is assigned with value '0' (user oriented notification message)
  - The terminal SHALL assume that attribute 'eventType' is assigned with value '1' (emergency notification)
  - The terminal SHALL assume that element 'Title' is present and expressed possibly in multiple languages.
  - The terminal SHALL assume that element 'Description' is present and expressed possibly in multiple languages.
  - The terminal SHALL assume that element 'PresentationType' is assigned with value '0' (high-priority notification messages)
  - The terminal MAY skip all the other elements and attributes in the Notification Message.

5.14.7 Guidelines for MediaInformation usage

As rich media presentation technologies are not specified in BCAST 1.0, the rendering of the Notification message is implementation specific. However the following section provides basic guidelines on the use of MediaInformation elements provide a consistent layout for the Notification messages. If required by service providers the guidelines below can be replaced with proprietary mechanisms.

- If multiple instances of Description element are present then the terminal SHOULD select and display the text of the Description based on the Terminal language setting.
- The Description element SHOULD always be set when using MediaInformation.
- Either the DeliverySession element or the AlternativeURI element SHOULD be set to indicate the delivery session providing the content to be used for MediaInformation.

- If the Picture element is set in the MediaInformation, the Picture element SHOULD be displayed first with the text contained in the Description element being displayed under the Picture element.
- If the Audio element is set in the MediaInformation, the Audio element SHOULD be played in conjunction with the text contained in the Description element being displayed.
- If both the Picture and the Audio elements are set in the MediaInformation then two guidelines above SHOULD be met.
- Video element in the MediaInformation MAY be supported by the terminal and if the Video element is supported by the terminal, the following guidelines apply:
  - If the Video element is set in the MediaInformation, the Video element SHOULD be displayed first with the text contained in the Description element being displayed under the Video element.
  - If both the Video and the Picture elements are set in the MediaInformation then only the Video element SHOULD be displayed.
  - If both the Video and the Audio elements are set in the MediaInformation then only the Video element SHOULD be displayed.

To support the possibility of multiple cardinalities of Picture and Audio elements in future release of BCAST 1.1 the Terminals supporting BCAST 1.0 SHOULD be able to parse and only select the first Picture or Audio element even if there are multiple instances of Picture and Audio elements in future BCAST 1.1 Notifications.

5.14.8 Extensibility placeholders for future usage of Notification

For the purpose of extending the usage of the Notification message in releases following the BCAST 1.0 specification while maximizing backward compatibility with 1.0 terminals, extensibility placeholders have been defined. These contains a wildcard schema component in which new attributes and elements can be inserted. Currently defined placeholders are:

- the ‘ServiceGuide’ element in the Notification message (see section 5.14.3)

Extensions targeting the extensibility placeholders SHALL NOT be defined within the XML namespaces applicable to BCAST 1.0.

The Terminal SHALL ignore Notification messages with values of the ‘eventType’ or ‘notificationType’ attributes that it does not support.

5.14.8.1 The ServiceGuide placeholder

The ‘ServiceGuide’ element in the Notification message is the placeholder for future extensions related to the Service Guide. It is of type ‘ServiceGuideType’, that is defined as an element containing the following wildcard:

```
<xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
```

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-BCAST_Services-V1_0-20090212-A	12 Feb 2009	pproved by TP TP ref# OMA-TP-2009-0071-INP_BCAST_V1_0_ERP_for_Notification_and_Final_Approval

Appendix B. Examples on Realizing Interactive Services (Informative)

Editor's note: this section may contain a walk-through for selected services that clarifies how the service can be generated, managed, and delivered, end-to-end.

B.1 Use of MMS Template for Service Interaction (Informative)

This section describes an example on how to use MMS Message Template for Service Interaction.

B.1.1 Retrieving the MMS Message Template

MMS Message Template can be broadcasted, as similar as other Service Interaction methods such as SMIL, XHTML MP etc.. In this case the files constructing MMS Message Template are concatenated in one GZIP and broadcasted within the file broadcast. The name and the MIME-type of each file are given in InteractivityMediaDocument (See Appendix.C for example).

MMS Message Template can also be retrieved from MMS. In this case the service provider or directly the operator author the MMS Template containing the MTD (MMS Message Template Definition), i.e. the template wizard toward the service. The template and some contents are embedded within a MMS Message with Multipart/Related or Multipart/Mixed format. The name and the MIME-type of each file are given in a header of the each part in Multipart Message. This MMS Message is send to the terminal whose users are registered to use Service Interaction.

The terminal will extract the files before the time when MMS Message Template is used in Service Interaction.

B.1.2 Launching MMS Message Template Client and creating Multimedia Message

After MMS Message Template retrieval, the terminal launches MMS Message Template Client with MMS Message Template. This section describes two cases for MMS Message Template use.

B.1.2.1 Use case: Voting

The first use case is Voting, for example, to vote 'who will win the game of the TV program'. In this case, MMS Message Template will have the following files:

- Message Template Definition (MTD) : using text-editor to input the name of the winner (shown below)

- MMS Presentation Part (SMIL)

- Media Objects (Text, Image)

Voting-sample.mtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mmstemplate PUBLIC "-//OMA//DTD MTD 1.3//EN"
"http://www.openmobilealliance.org/MMS/MTD/1.3/DTD/mtd13.dtd">
<mmstemplate xmlns="http://www.openmobilealliance.org/2004/mtd">
   <head>
     <title>Vote the winner</title>
     <description>MTD sample code for BCAST Service Interaction</description>
     <date>2005-10-10</date>
     <version>1.00</version>
     <author>John Doe</author>
   </head>
   <body>
     <message>
        <to-header editable="false">1677721664</to-header>
        <subject-header>Vote the winner</subject-header>
     </message>
     <wizard>
        <step guide="Please input the name of the winner " app="text-input" target-name="name.txt"
target-type="text/plain" required="true"/>
     </wizard>
   </body>
</mmstemplate>
```

Table 50: MMS Template Example for Voting

MMS Message Template Client could display the following text input screen.

Note: MMS Message Template only specifies the input method. It does not specify the screen flow and how to construct text input screen. The appearance of the input screen will depend on the implementation of the client.

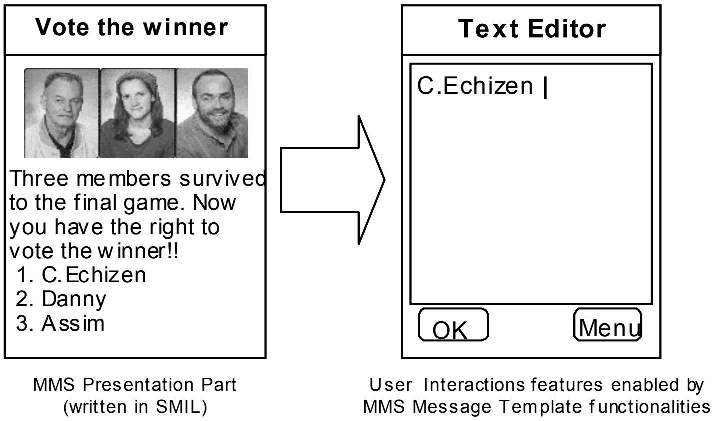

Vote the winner
Three members survived to the final game. Now you have the right to vote the winner!!
1. C.Echizen
2. Danny
3. Assim
Text Editor
C.Echizen |
OK
Menu
MMS Presentation Part (written in SMIL)
User Interactions features enabled by MMS Message Template functionalities

Figure 4: The screen flow of Voting Template

B.1.2.2 Use case: Viewer’s Contribution

The second use case is Viewer's Contribution, for example, to send the viewer's pet boast to the TV program.

In this case, MMS Message Template will have the following files:

- Message Template Definition (MTD) :

  description that uses still camera to take a photo of the pet, and text editor to input the comment (shown below).

- MMS Presentation Part (SMIL)

- Media Objects (Text, Image)

Contribution-sample.mtd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mmstemplate PUBLIC "-//OMA//DTD MTD 1.3//EN"
"http://www.openmobilealliance.org/MMS/MTD/1.3/DTD/mtd13.dtd">
<mmstemplate xmlns="http://www.openmobilealliance.org/2004/mtd">
   <head>
     <title>Boast of my pet</title>
     <description>MTD sample code for BCAST Service Interaction</description>
     <date>2005-10-10</date>
     <version>1.00</version>
     <author>John Doe</author>
   </head>
   <body>
     <message>
```

```
        <to-header editable="false">1677721664</to-header>
        <subject-header>Show your pet off</subject-header>
     </message>
     <wizard>
        <step guide="Please take the picture of your pet" app="still-camera" target-name="photo.jpg"
target-type="image/jpg" required="true"/>
        <step guide="Please input your comment" app="text-input" target-name="comment.txt" target-
type="text/plain" required="true"/>
     </wizard>
   </body>
</mmstemplate>
```

Table 51: MMS Template Example for User Feedback

MMS Message Template Client will show the multiple input screens. The first screen will be the camera application and next one will be text editor. The example of input screens could be figured as follows:

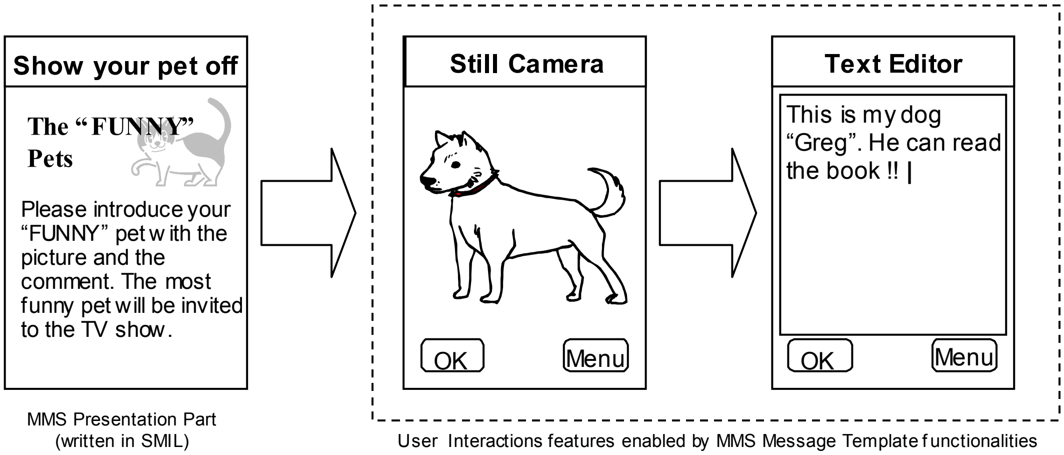

Show your pet off
The "FUNNY" Pets
Please introduce your "FUNNY" pet with the picture and the comment. The most funny pet will be invited to the TV show.
Still Camera
OK
Menu
Text Editor
This is my dog "Greg". He can read the book !! |
OK
Menu
MMS Presentation Part (written in SMIL)
User Interactions features enabled by MMS Message Template functionalities

Figure 5: The screen flow of Viewer's Contribution Template

B.1.3 Sending the Interaction Message

The Resulting MM created by MMS Message Template Client will be sent to BCAST Service Application via MMS through SI-8.

Appendix C. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

Note 1: References refer to this specification unless otherwise noted.

Note 2: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (Broadcast Distribution System), may overrule or adapt requirements from this SCR or provide additional requirements

C.1 SCR for BCAST Client

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-C-001	Terminal with access to interaction channel	general	O	BCAST-SERVICES-C-011 AND BCAST-SERVICES-C-012 AND BCAST-SERVICES-C-013 AND BCAST-SERVICES-C-017 AND BCAST-SERVICES-C-018 AND BCAST-SERVICES-C-019 AND BCAST-SERVICES-C-020 AND BCAST-SERVICES-C-025 AND BCAST-NT-C-003 AND BCAST-NT-C-005
BCAST-SERVICES-C-002	Terminal with access to interaction channel and support for Service and/or Content Protection	general, [BCAST10-ServContProt]	O	BCAST-SERVICES-C-006 AND BCAST-SERVICES-C-007 AND BCAST-SERVICES-C-008
BCAST-SERVICES-C-003	Terminal supporting SMS	general	O	BCAST-SERVICES-C-014
BCAST-SERVICES-C-004	Terminal supporting MMS	general	O	BCAST-SERVICES-C-015
BCAST-SERVICES-C-005	Terminal supporting Voice call	general	O	BCAST-SERVICES-C-016
BCAST-SERVICES-C-006	Service Provisioning	Section 5.1	O	
BCAST-SERVICES-C-007	HTTP POST for service provisioning	Section 5.1.1	O	
BCAST-SERVICES-C-008	Provisioning Messages	Section 5.1	O	BCAST-SERVICES-C-009
BCAST-SERVICES-C-009	GZIP compression of Provisioning Messages	Section 5.1.7	O	
BCAST-SERVICES-C-010	Web-based Service Provisioning	Section 5.1.8	O	
BCAST-SERVICES-C-011	Terminal Provisioning using OMA DM	Sections 5.2, 5.2.2	O	
BCAST-SERVICES-C-012	Reception of terminal provisioning messages and update of the parameters included in the terminal provisioning messages	Section 5.2	O	

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-C-013	Service interaction using IP, TCP, HTTP	Section 5.3.1	O	
BCAST-SERVICES-C-014	Service interaction using SMS	Sections 5.3.1, 5.3.6.1.2., 5.3.6.1.3	O	
BCAST-SERVICES-C-015	Service interaction using MMS	Sections 5.3.1., 5.3.6.1.2	O	
BCAST-SERVICES-C-016	Service interaction using Voice Call	Section 5.3.6.1.2	O	
BCAST-SERVICES-C-017	Interactive retrieval of SG	Section 5.3.2	O	
BCAST-SERVICES-C-018	Interactive retrieval of Service Guide related information	Section 5.3.3	O	
BCAST-SERVICES-C-019	Reception of InteractivityMedia documents over broadcast file distribution	Section 5.3.6.1, 5.3.6.2	O	
BCAST-SERVICES-C-020	Retrieval of InteractivityMedia documents and associated files over interaction channel	Section 5.3.6.1, 5.3.6.3	O	
BCAST-SERVICES-C-021	Rendering of InteractivityMedia objects	Section 5.3.6.1	M	
BCAST-SERVICES-C-022	Acquisition and rendering of the media objects attached to the InteractivityMedia document without interrupting the acquisition and rendering of the ‘regular’ broadcast media stream	Section 5.3.6.1.3	M	
BCAST-SERVICES-C-023	Description and evaluation of end user preferences	Section 5.4	O	BCAST-SERVICES-C-024
BCAST-SERVICES-C-024	Format of end user preference description	Section 5.4.2	O	
BCAST-SERVICES-C-025	Broadcast Roaming	Section 5.7	O	BCAST-SERVICES-C-026
BCAST-SERVICES-C-026	Format of roaming messages	Sections 5.7.1	O	
BCAST-SERVICES-C-027	Support of Location Information	Section 5.8	O	BCAST-SERVICES-C-028 OR BCAST-SERVICES-C-029 OR BCAST-SERVICES-C-030
BCAST-SERVICES-C-028	Support of Location Information in OMA MLP format	Section 5.8	O	
BCAST-SERVICES-C-	Support of Location	Section 5.8	O	

Item	Function	Reference	Status	Requirement
029	Information in zip code format			
BCAST-SERVICES-C-030	Support of Location Information in BDS-specific cell_id format	Section 5.8	O	
BCAST-SERVICES-C-031	XML formatting rules for signalling	Section 5.9	M	
BCAST-SERVICES-C-032	3GPP Timed Text for Subtitling and Closed Captions	Section 5.13	O	

C.2 SCR for BCAST Service Application (BSA)

The BSA is an entity in the OMA BCAST Architecture, see [BCAST10-Architecture] Fig. 3.

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-BSA-001	Service interaction using one or several of: IP, TCP, HTTP, SMS, IPSEC, UDP, MMS, WAP, HTTPS based on SSL 3.0 [SSL30] and TLS 1.0 [RFC 2246], SIP/IMS	Section 5.3.1	O	
BCAST-SERVICES-BSA-002	Support for Interactivity MediaDocument format and delivery	Section 5.3.6.1.2	O	

C.3 SCR for BCAST Service Distribution/Adaptation (BSDA)

The BSDA is an entity in the OMA BCAST Architecture, see [BCAST10-Architecture] Fig. 3.

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-BSDA-001	Description and evaluation of end user preferences	Section 5.4	O	BCAST-SERVICES-BSDA-002
BCAST-SERVICES-BSDA-002	Format of end user preference description	Section 5.4.1	O	
BCAST-SERVICES-BSDA-003	Use of Location Information	Section 5.8	O	
BCAST-SERVICES-BSDA-004	Use of Location Information in OMA MLP format	Section 5.8	O	
BCAST-SERVICES-BSDA-005	Use of Location Information in zip code format	Section 5.8	O	
BCAST-SERVICES-BSDA-006	Use of Location Information in BDS-specific cell_id format	Section 5.8	O	
BCAST-SERVICES-BSDA-007	XML formatting rules for signalling	Section 5.9	M	

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-BSDA-008	Subtitling and Closed Captions	Section 5.13	O	

C.4 SCR for BCAST Subscription Management (BSM)

The BSM is an entity in the OMA BCAST Architecture, see [BCAST10-Architecture] Fig. 3.

Item	Function	Reference	Status	Requirement
BCAST-SERVICES-BSM-001	Service Provisioning	Section 5.1	M	
BCAST-SERVICES-BSM-002	HTTP POST for service provisioning	Section 5.1.1	M	
BCAST-SERVICES-BSM-003	GZIP compression of Provisioning Messages	Section 5.1.7	M	
BCAST-SERVICES-BSM-004	Web-based Service Provisioning	Section 5.1.8	O	
BCAST-SERVICES-BSM-005	Terminal Provisioning using OMA DM	Section 5.2	M	
BCAST-SERVICES-BSM-006	Delivery of OMA DM messages through Interaction Channel using DM mechanism	Section 5.2.4	M	
BCAST-SERVICES-BSM-007	Broadcast Roaming	Section 5.7	O	BCAST-SERVICES-BSM-008
BCAST-SERVICES-BSM-008	Format of roaming messages	Sections 5.7.1, 5.7.2	O	
BCAST-SERVICES-BSM-009	XML formatting rules for signalling	Section 5.9	M	
BCAST-SERVICES-BSM-010	Protocol stack for message exchanges between BSMs	Section 7.2.1	M	

C.5 SCR for BCAST Notification Client (NTC)

Item	Function	Reference	Status	Requirement
BCAST-NT-C-001	Support for the signalling of the availability and access to generic notifications through the SGDD.	Sections 5.14.1.1.1, [BCAST10-SG] 5.4.2.5	O	
BCAST-NT-C-002	Support for the signalling of the availability and access to service-specific notifications through 'Access' fragment.	Sections 5.14.1.2.1, [BCAST10-SG] 5.1.2.4	O	
BCAST-NT-C-003	Support for subscribing to notifications by sending a Notification	Section 5.14.4.2.1	O	

Item	Function	Reference	Status	Requirement
	Request to NTG			
BCAST-NT-C-004	Support for user-oriented notification request message format	Section 5.14.4.2.1	O	BCAST-NT-C-003
BCAST-NT-C-005	Support for polling to notifications over Interaction Channel	Section 5.14.4.3	O	

C.6 SCR for BCAST Notification Distribution Adaptation (NTDA)

Item	Function	Reference	Status	Requirement
BCAST-NT-DA-001	Support for the signalling of the availability and access to generic notifications through the SGDD.	Sections 5.14.1.1.1, [BCAST10-SG] 5.4.2.5	O	
BCAST-NT-DA-002	Support for the signalling of the availability and access to service-specific notifications through the 'Access' fragment.	Sections 5.14.1.2.1, [BCAST10-SG] 5.1.2.4	O	
BCAST-NT-DA-003	Notification back-end interface exposed	Section 5.14.5.1	O	BCAST-NT-DA-004 AND BCAST-NT-DA-005
BCAST-NT-DA-004	Support back-end interface for notification function	Section 5.14.5.1	O	
BCAST-NT-DA-005	Support the back-end message for notification	Section 5.14.5.2	O	
BCAST-NT-DA-006	Backend interface SG-4 exposed in implementation	Section 5.14.5.1	O	BCAST-NT-DA-007
BCAST-NT-DA-007	Support backend interface SG-4 for SG function	Section 5.14.5.1	O	(BCAST-NT-DA-008 OR BCAST-NT-DA-009) AND BCAST-NT-DA-010 AND (BCAST-NT-DA-011 OR BCAST-NT-DA-012) AND BCAST-NT-DA-013 AND BCAST-NT-DA-005
BCAST-NT-DA-008	Support IPv4	Section 5.14.5.1	O	
BCAST-NT-DA-009	Support IPv6	Section 5.14.5.1	O	
BCAST-NT-DA-010	Support TCP	Section 5.14.5.1	O	

Item	Function	Reference	Status	Requirement
BCAST-NT-DA-011	Support HTTP1.1	Section 5.14.5.1	O	
BCAST-NT-DA-012	Support HTTPS	Section 5.14.5.1	O	
BCAST-NT-DA-013	SG backend messages for content delivery	Section 5.14.5.1	O	

Appendix D. <MediaObjectSet> examples (Informative)

This appendix provides illustrative examples of <MediaObjectSet> elements present in InteractivityMedia documents. The external file (GZIP archive or single media file part) is not given.

D.1 XHTML MP bundle

Example of an XHTML MP bundle containing two XHTML pages, one picture and one external WAP CSS stylesheet:

```
<MediaObjectSet
     RelativePreference="5"
     Content-Type="application/x-gzip"
     Content-Location="http://www.bcast.com/purchaseme.gz"
     xml:lang="en-UK"
 >
     <Object
         Content-Type="application/vnd.wap.xhtml+xml"
         Content-Location="index.html"
         Start="true" />
     <Object
         Content-Type="application/vnd.wap.xhtml+xml"
         Content-Location="other.html" />
     <Object
         Content-Type="text/css"
         Content-Location="./style/style.css" />
     <Object
         Content-Type="image/gif"
         Content-Location="./images/background.gif" />
     <Description xml:lang="en">Purchase me</Description>
     <Description xml:lang="fr">Achetez moi</Description>
 </MediaObjectSet>
```

File structure after deflation would be :

index.html

other.html

/style/style.css

/images/background.gif

with ‘index.html’ typically containing the following links :

<link rel= "stylesheet" href= "./style/style.css" />

<a href = "other.html"> Click to see next page </a>

<img src = "./image/background.gif" />

D.2 MMS Message Template bundle

Example of an MMS Message Template bundle containing one MMS Template Definition, one SMIL and one text part and one picture:

```
<MediaObjectSet
     RelativePreference="10"
     Content-Type="application/x-gzip"
     Content-Location="http://www.bcast.com/votenow.gz"
 >
     <Object
         Content-Type="application/vnd.omammsg-mtd+xml"
         Content-Location="votenow.mtd" />
         Start="true" />
     <Object
         Content-Type="application/smil"
         Content-Location="presentation.smil" />
     <Object
         Content-Type="image/png"
         Content-Location="title.png" />
     <Object
         Content-Type="text/plain"
         Content-Location="vote.txt" />
 </MediaObjectSet>
```

File structure after deflation would be :

votenow.mtd

presentation.smil

title.png

vote.txt

with ‘presentation.smil’ typically containing the following links :

<img src = "title.png" />

<text src = "vote.txt" />

D.3 SMIL bundle

Example of a SMIL bundle containing one XHTML MP Rich Text and one Audio file :

```
<MediaObjectSet
     RelativePreference="8"
     Content-Type="application/x-gzip"
     Content-Location="http://www.bcast.com/inputtimeout.gz"
 >
     <Object
         Content-Type="application/smil"
         Content-Location="presentation.smil"
         Start="true" />
     <Object
         Content-Type="application/vnd.wap.xhtml+xml"
         Content-Location="farewell.html" />
     <Object
         Content-Type="audio/3gpp"
         Content-Location="./audio/symphony.3gp" />
 </MediaObjectSet>
```

File structure after deflation would be :

presentation.smil

```
farewell.html

/audio/symphony.3gp
```

with ‘presentation.smil’ typically containing the following links :

```
<text src= "farewell.html" type="application/vnd.wap.xhtml+xml" />

<audio src= "./audio/symphony.3gp" type="audio/3gpp" />
```

Appendix E. Walk-through of Broadcast Roaming (Informative)

This appendix illustrates how the Broadcast Roaming is achieved through the use of core functionalities of BCAST 1.0. This informative explanation of Broadcast Roaming is presented as a walk-through mainly from the terminal point of view.

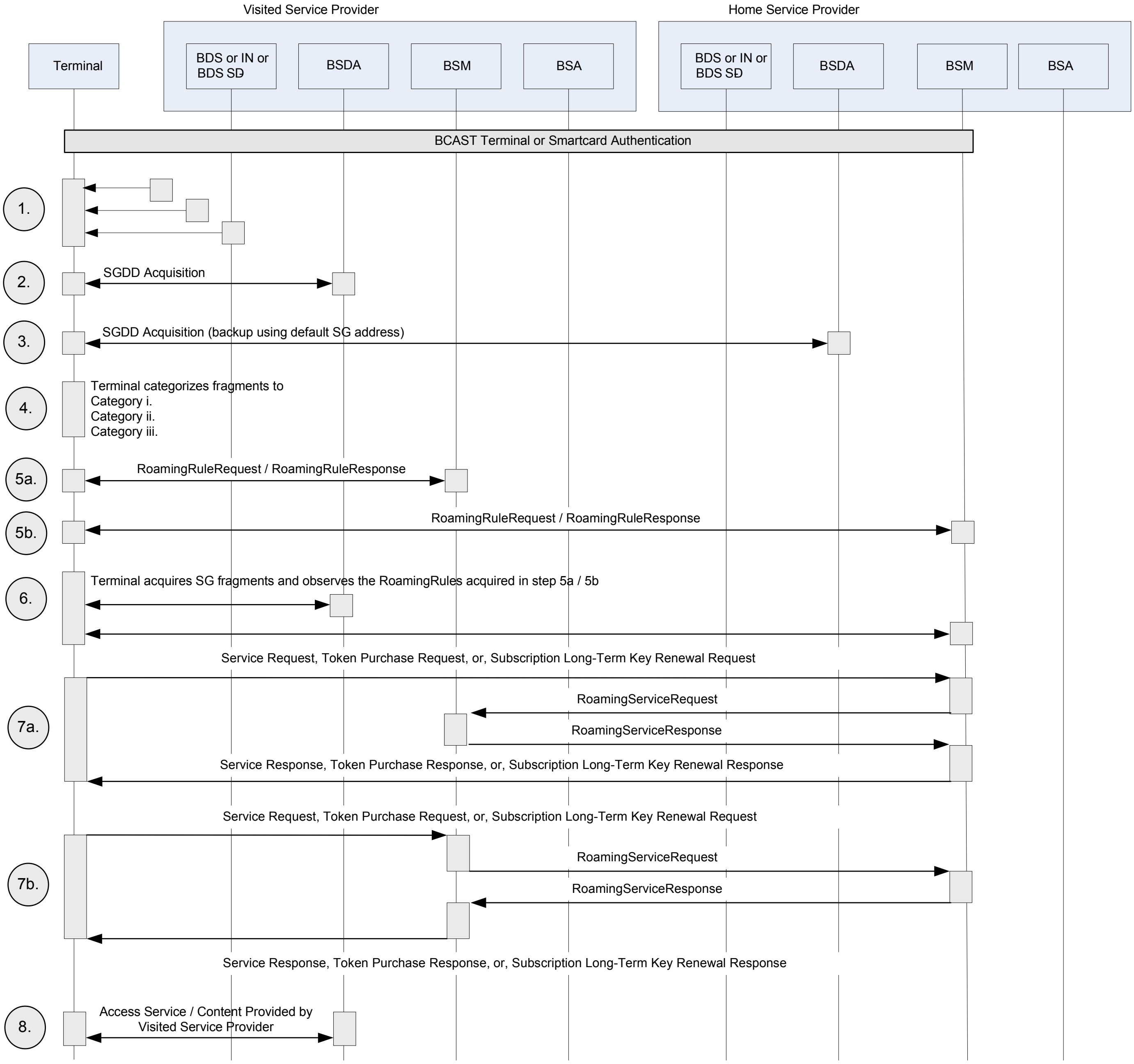

Visited Service Provider
Home Service Provider
Terminal
BDS or IN or BDS SD
BSDA
BSM
BSA
BDS or IN or BDS SD
BSDA
BSM
BSA
BCAST Terminal or Smartcard Authentication
1.
2.
SGDD Acquisition
3.
SGDD Acquisition (backup using default SG address)
4.
Terminal categorizes fragments to
Category i.
Category ii.
Category iii.
5a.
RoamingRuleRequest / RoamingRuleResponse
5b.
RoamingRuleRequest / RoamingRuleResponse
6.
Terminal acquires SG fragments and observes the RoamingRules acquired in step 5a / 5b
Service Request, Token Purchase Request, or, Subscription Long-Term Key Renewal Request
7a.
RoamingServiceRequest
RoamingServiceResponse
Service Response, Token Purchase Response, or, Subscription Long-Term Key Renewal Response
Service Request, Token Purchase Request, or, Subscription Long-Term Key Renewal Request
7b.
RoamingServiceRequest
RoamingServiceResponse
Service Response, Token Purchase Response, or, Subscription Long-Term Key Renewal Response
8.
Access Service / Content Provided by Visited Service Provider

Figure 6: Informative Example of Broadcast Roaming

The walk-through below is illustrated as flow chart in Figure D.1.

1. Terminal scans or otherwise detects available Broadcast Distribution Systems (BDS).
2. Terminal attempts to perform Service Guide discovery bootstrap to locate entry point to BCAST Service Guide on all or any of the detected BDSes. Upon successful completion of bootstrap procedure, the Terminal acquires the

entry point to BCAST Service Guide over the respective bearer. Consequently, the Terminal acquires SGDDs either by receiving or by retrieving those.

3. In case Terminal fails to perform bootstrap and to locate the entry point to BCAST Service Guide over all the detected BDSes, the Terminal attempts to retrieve SGDDs using the entry point as provisioned in the Terminal (defined by Management Object "<X>/SGServerAddress").
4. Once the Terminal acquires SGDDs, the Terminal looks for BSMSelector elements and BSMFilterCodes within those elements in the SGDD. Together with that information and the terminal's affiliated BSM(s) which are represented within the Terminal as Management Objects with identifier '<X>/BSMFilterCode', the Terminal categorizes all the fragments declared in the SGDD into three categories:

    i. Fragments that are associated with a BSMFilterCode (within BSMSelector), which matches at least one of the BSMFilterCodes associated with Home Mobile Broadcast Service Provider (<X>/ BSMFilterCode/IsHomeBSM == true). Terminal can use, interpret and render the information contained in these fragments without restrictions.

    ii. Fragments that are associated with a BSMFilterCode (within BSMSelector), which does not match with any of the BSMFilterCodes associated with the terminal or match BSMFilterCodes associated with Visited Mobile Broadcast Service Provider (<X>/ BSMFilterCode/IsHomeBSM == false). Terminal can render, interpret and handle the fragments according to RoamingRules associated with this BSMSelector. BSMSelector and the associated RoamingRules are identified by the attribute "Id" present within the BSMSelector as well as in RoamingRules. In the RoamingRules have been provisioned using BCAST Terminal Provisioning, the rules are associated with each BSMFilterCode, under <X>/ BSMFilterCode/RoamingRule.

    iii. Fragments that are not associated with any BSMFilterCode (no BSMSelector).

    - In case Terminal has no Management Objects with identifier '<X>/ BSMFilterCode' present, the Terminal can use, interpret and render the information contained in these fragments without restrictions.
    - In case Terminal has at least one Management Object with identifier '<X>/ BSMFilterCode' present, the Terminal will determine behaviour according to Management Objects with identifier '<X>/Roaming/IgnoreUnIdentifiedBSM'.
        - If the Management Objects with identifier '<X>/Roaming/ IgnoreUnIdentifiedBSM' is set with value "true" the Terminal cannot use, interpret and render the information contained in these fragments at all.
        - If the Management Objects with identifier '<X>/Roaming/ IgnoreUnIdentifiedBSM' is set with value "false" the Terminal can use, interpret and render the information contained in these fragments without restrictions.
        - If the Management Objects with identifier '<X>/Roaming/IgnoreUnIdentifiedBSM' is not present, the Terminal assumes that the value of this Management Object is "true" and the Terminal cannot use, interpret and render the information contained in these fragment at all.

5. If the terminal wants to render, interpret and handle the fragments in category (ii.) above, it needs to acquire the RoamingRules related to the BSMSelector in question. There are three ways to achieve this.

    a. The Terminal fetches the RoamingRules from Visited BSM. For that, the BSMSelector contains attribute "RoamingRuleRequestAddress" to which the Terminal can address the RoamingRuleRequest. As a response of to the RoamingRuleRequest the Terminal will receive RoamingRuleResponse which contains the RoamingRules associated with the BSMSelector.

    b. The Terminal fetches the RoamingRules from Home BSM. This happens if the BSMSelector does not have "RoamingRuleRequestAddress" present, OR, if the Terminal has Management Object "<X>/Roaming/ForceHomeRoamingRuleRequestAddress" present and set to "true". In these cases the Terminal sends the RoamingRuleRequest to "<X>/Roaming/HomeRoamingRuleRequestAddress". As a response of the RoamingRuleRequest the Terminal will receive RoamingRuleResponse which contains the RoamingRules associated with the BSMSelector.

c. The RoamingRules were originally provided as a part of BSMSelector (not illustrated in figure D.1)

6. The Terminal acquires Service Guide fragments. It interprets, handles and renders the fragments according to RoamingRules. Consequently the Terminal uses the Service Guide fragments to perform subscriptions to services and content, and to access services and content described by the Service Guide.

7. Depending on the value of Management Object "<X>/Roaming/UseVisitedServiceProvisioningMode" the terminal determines whether to initiate the service provisioning request to Visited BSM or to Home BSM. Then the terminal sends the message to either Visited BSM or Home BSM. The receiving system determines from the requested GlobalPurchaseItemId and included UserID whether the request is about roaming. Two cases for this exist: either the Terminal sends the Service Request message to its Home BSM or to the Visited BSM.

    a. In the former case Home BSM detects that one of its terminal is requesting PurchaseItem served by another BSM. If the Home BSM wants to allow terminal to access the PurchaseItem, the Home BSM goes ahead and sends RoamingServiceRequest to the Visited BSM. Visited BSM answers with RoamingServiceResponse. In case the response allows roaming, then the Home BSM sends a successful ServiceResponse to the terminal. This leads to a subsequent LTKM delivery (push LTKM with Smartcard Profile or Trigger with DRM Profile). The LTKM acquisition is not shown in the diagram as it is a Service & Content Protection procedure.

    b. In the latter case Visited BSM detects that a terminal that is not one of the terminals affiliated with this BSM is requesting PurchaseItem served by this BSM. The Visited BSM consequently sends RoamingServiceRequest to the Home BSM of the terminal. Home BSM answers with RoamingServiceResponse. In case the response allows roaming, then the Visited BSM sends a successful ServiceResponse to the terminal. This leads to a subsequent LTKM delivery (push LTKM with Smartcard Profile or Trigger with DRM Profile). The LTKM acquisition is not shown in the diagram as it is a Service & Content Protection procedure.

    Upon successful RoamingProvisioning, the Terminal is granted right to purchase and/or subscribe to the PurchaseItem it requested.

Additionally, iin case the Terminal decides to request Long Term Key or to renew Long Term Key associated with a subscription, the Terminal sends either 'LTKM Request' or 'Subscription LTKM Renewal Request'. Two cases for this exist: either the Terminal sends the message to its Home BSM or to the Visited BSM.

a. In the former case Home BSM detects that one of its terminal is requesting LTKM or renewal of LTKM associated with PurchaseItem served by another BSM. If the Home BSM wants to allow terminal to access the LTKM, the Home BSM goes ahead and sends RoamingAuthorizationRequest to the Visited BSM.

b. In the latter case Visited BSM detects that a terminal that is not one of the terminals affiliated with this BSM is requesting LTKM or renewal of LTKM associated with PurchaseItem served by this BSM. The Visited BSM consequently sends RoamingAuthorizationRequest to the Home BSM of the terminal.

   Note: If step 8a or 8b follow 7a or 7b within a certain time frame, the authorization between home and visited BSM is not necessary.

8. The Terminal accesses service and/or content related to PurchaseItem, provided by Visited Service Provider.

Appendix F. BCAST Management Object

F.1 OMA BCAST Device Management general

BCAST MOs allow a device to present the configuration of the device in a standardized way, allowing a server to be able to bootstrap, retrieve and manage the configuration of a device (the parameters included in the MO).

The BCAST MO structure is formally defined in [BCAST10-DDF-BCAST-MO].

Note: for the semantics of 'Status' of each of the MO parameters see [DMACMO]. This information is repeated here for convenience.

- Definition:
  - The Status definition in the node definitions indicates if the Client must support that node or not.
  - If the Status is "Required" even though the node may not be present at that time, the Server can expect the Client to be able to support it.
  - If the Status is "Required" then the Client must support that node in the case the Client supports the parent node. If the status is "Optional" then this should be reflected in DDF file to show whether the node is supported.
  - When creating the status of an MO, the child may be Required, while the parent node may be Optional. This would mean that all those elements would be Optional, but in case the parent node is present, then those child nodes would be Required.
- Possible Values: The value of this parameter can be "Required" or "Optional".

F.2 OMA BCAST Management Object Tree

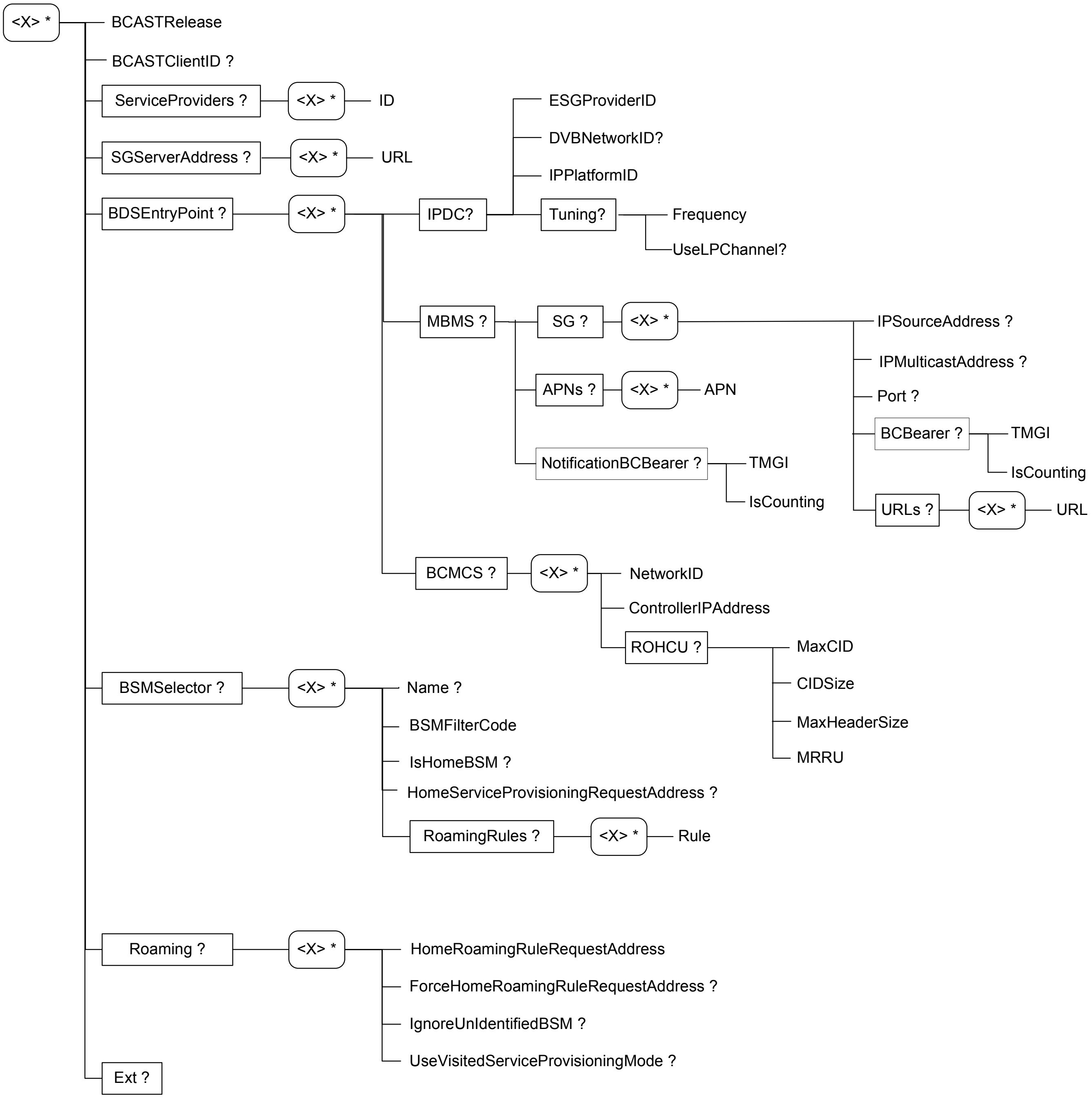


Figure 7: OMA BCAST Management Object Structure

Note: “?” means zero or one occurrences, “*” means zero or more occurrences. No symbol means one occurrence.

F.3 BCAST MO parameters

This section provides a description of the elements of the BCAST MO. Unless otherwise stated, BCAST terminals SHALL support the nodes defined below.

F.3.1 <X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for one or more BCAST Management Object root nodes. It is MANDATORY if the UE supports OMA BCAST.

F.3.2 <X>/BCASTRelease

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the BCAST release of the client. It is MANDATORY and MUST have the value “1.0” for this release.

F.3.3 <X>/BCASTClientID

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node contains the BCAST_Client_ID used by the Smartcard Profile as per [BCAST10-ServContProt].

F.3.4 <X>/ServiceProviders

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node acts as a container for a list of Service Provider identifiers.

F.3.5 <X>/ServiceProviders/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a list of Service Provider identifiers.

F.3.6 <X>/ServiceProviders/<X>/ID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the Service Provider identifier for the BCAST Service. It is e.g. used in the ‘serviceproviders’ field for protection signalling in SDP as per section 10.1.1 of [BCAST10-SrvContProt]. The value of this node MUST be in the form of a URI.

F.3.7 <X>/SGServerAddress

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains information about BCAST Service Guide Servers for the interactive mode. In case there are multiple servers present, the terminal MAY use any of them.

F.3.8 <X>/SGServerAddress/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node serves as a placeholder for a list of addresses of BCAST Service Guide Servers for the interactive mode.

F.3.9 <X>/SGServerAddress/<X>/URL

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the BCAST Service Guide server URL for the interactive mode.

F.3.10 <X>/BDSEntryPoint

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node contains information about the service entry points in the different BDSs. Possible children: IPDC, MBMS.

> It is RECOMMENDED to also include Add, Delete and Replace access types on the implementations, in order to support write access on the sub-nodes to provision the necessary sets of information.

F.3.11 <X>/BDSEntryPoint/<X>

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrMore	node	Get

This interior node acts as a placeholder for sets of BDS-specific information.

If more than one instance of this node are present, the terminal MAY use suitable means (like the reception quality or user selection) to choose the most appropriate one.

F.3.12 <X>/BDSEntryPoint/<X>/IPDC

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

For a terminal using IPDC as the BDS, it is necessary to provision some information how to tune the device to the DVB-H broadcast network and to discover the IP flows in it.

If this interior node is present, a terminal using the DVB-IPDC BDS SHOULD use this information to tune its receiver, to discover the IP flows which carry the service, and to resolve the actual Service Guide to use in a multi provider scenario.

This node acts as a container for all the BDS-specific information regarding IPDC over DVB-H. BCAST Terminals MAY support this node and its sub-nodes.

F.3.13 <X>/BDSEntryPoint/<X>/IPDC/Tuning

Status	Occurrence	Format	Min. Access Types

Optional	ZeroOrOne	node	Get

This interior node contains tuning parameters for the DVB-H receiver.

F.3.14 <X>/BDSEntryPoint/<X>/IPDC/Tuning/Frequency

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node carries the center frequency of the DVB-H channel to tune to.

The value representsthe frequency in kHz. This MUST be a decimal number and MUST fit within the range of a 32 bit unsigned integer.

F.3.15 <X>/BDSEntryPoint/<X>/IPDC/Tuning/UseLPChannel

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

DVB-H may use an optional hierarchical modulation mode in which case the receiver needs to make a selection between a “high priority” (HP) channel and a “low priority” (LP) channel.

This leaf node provides the information which is needed to tune to a hierarchically modulated DVB-H channel.

If present and **true**, the terminal SHALL use the LP channel in DVB-H hierarchical modulation. If not present or **false**, the terminal SHALL use the HP channel in DVB-H hierarchical modulation or assume that no hierarchical modulation is used.

F.3.16 <X>/BDSEntryPoint/<X>/IPDC/IPPlatformID

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

DVB uses the concept of IP platforms to disambiguate the IP address ranges of several sources of IP traffic sharing a DVB channel. For a DVB-H terminal, the IP platform ID is required as side information to discover the IP flows.

According to [ETSI 102 470], section 4.2, an IP platform ID value is either registered with DVB in which case it is globally unique, or it is scoped to the network ID (see next section).

This leaf node provides

> the IP Platform ID. This node MUST contain a decimal number and MUST fit within the range of a 24 bit unsigned integer.

F.3.17 <X>/BDSEntryPoint/<X>/IPDC/DVBNetworkID

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

There are cases where the IP platform ID is not globally unique but scoped to a DVB network ID which is registered with DVB.

This leaf node provides the network ID. It SHALL be present only if the IP platform ID is not globally unique according to [ETSI 102 470], section 4.2.

This node MUST contain a decimal number and MUST fit within the range of a 16 bit unsigned integer.

F.3.18 <X>/BDSEntryPoint/<X>/IPDC/ESGProviderID

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

In a DVB-IPDC deployment, multiple service providers can share a DVB-H channel. The Service Guide bootstrap session can therefore contain multiple Service Guides (one per service provider and IP platform). To select and receive a service guide via the DVB-IPDC BDS, the terminal needs to know the ID of the service guide provider to be used.

This leaf node provides

Service Guide Provider ID for SG bootstrapping. This node MUST contain a decimal number and MUST fit within the range of a 16 bit unsigned integer.

F.3.19 <X>/BDSEntryPoint/<X>/MBMS

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior acts as a container for all the BDS-specific information regarding MBMS. BCAST Terminals MAY support this node and its sub-nodes.

F.3.20 X>/BDSEntryPoint/<X>/MBMS/SG

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains bootstrap parameters for SG reception over MBMS broadcast bearer or SG retrieval over MBMS unicast bearer.

F.3.21 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for sets of bootstrap parameters for SG reception over MBMS broadcast bearer or SG retrieval over MBMS unicast bearer.

F.3.22 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/IPSourceAddress

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node contains the IP Source Address of the SG delivery session for a broadcasted SG.

F.3.23 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/IPMulticastAddress

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node contains the IP Multicast Address of the SG Announcement Channelfor a broadcasted SG.

F.3.24 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/Port

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	int	Get

This leaf node contains the port number for a broadcasted SG. The value MUST be a decimal number and MUST fit within the range of a 16 bit unsigned integer.

F.3.25 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/BCBearer

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node acts as a placeholder for sets of MBMS bearer parameter used for reception of a broadcasted SG (both SG Announcement Channel and SG Delivery Channel). This node SHALL be present in case the mode of the MBMS bearer is Broadcast Mode, and SHALL be absent otherwise.

F.3.26 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/BCBearer/TMGI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the Temporary Mobile Group Identity (TMGI) for a broadcasted SG as defined in [3GPP TS 23.003]. An MBMS Bearer service is uniquely identified by the TMGI. The value is encoded as a decimal number which in hexadecimal form represent octets 3 to 8 of the TMGI information element structure defined in [3GPP TS 24.008]. Octet 3 is the most significant octet.

F.3.27 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/BCBearer/IsCounting

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

This leaf node contains the information element MBMS Counting Information as defined in [3GPP TS 25.413]. It indicates whether the RAN level counting procedures is applicable or not for the MBMS broadcast mode.

The value **true** corresponds to the information element value of “counting” and the value **false** corresponds to the information element value “not counting”.

It is OPTIONAL for the terminal to act on to the information provided in this leaf node, e.g., if it has received the counting information out-of-band of the BCAST MO.

F.3.28 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/URLs

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains a list of URLs where an SDP describing the delivery session of a broadcasted SG can be fetched.

F.3.29 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/URLs/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a list of URLs where an SDP describing the Announcement Channel of a broadcasted SG can be fetched.

F.3.30 <X>/BDSEntryPoint/<X>/MBMS/SG/<X>/URLs/<X>/URL

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the URL where an SDP describing the Announcement Channel of a broadcasted SG can be fetched.

F.3.31 <X>/BDSEntryPoint/<X>/MBMS/APNs

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains a list of URIs of usable Access Point Names (APN).

F.3.32 <X>/BDSEntryPoint/<X>/MBMS/APNs/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a list of URIs of usable Access Point Names (APN).

F.3.33 <X>/BDSEntryPoint/<X>/MBMS/APNs/<X>/APN

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the URI of a usable Access Point Name (APN). An MBMS bearer is identified by IP multicast address and APN.

F.3.34 <X>/BDSEntryPoint/<X>/MBMS/NotificationBCBearer

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node acts as a placeholder for sets of MBMS bearer parameter used for the Notification Function. This node SHALL be present in case the mode of the MBMS bearer is Broadcast Mode, and SHALL be absent otherwise.

F.3.35 <X>/BDSEntryPoint/<X>/MBMS/NotificationBCBearer/TMGI

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the Temporary Mobile Group Identity (TMGI) for a broadcasted SG as defined in [3GPP TS 23.003]. An MBMS Bearer service is uniquely identified by the TMGI. The value is encoded as a decimal number which in hexadecimal form represent octets 3 to 8 of the TMGI information element structure defined in [3GPP TS 24.008]. Octet 3 is the most significant octet.

F.3.36 <X>/BDSEntryPoint/<X>/MBMS/NotificationBCBearer/IsCounting

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

This leaf node contains the information element MBMS Counting Information as defined in [3GPP TS 25.413]. It indicates whether the RAN level counting procedures is applicable or not for the MBMS broadcast mode.

The value **true** corresponds to the information element value of “counting” and the value **false** corresponds to the information element value “not counting”.

It is OPTIONAL for the terminal to act on to the information provided in this leaf node, e.g., if it has received the counting information out-of-band of the BCAST MO.

F.3.37 <X>/BDSEntryPoint/<X>/BCMCS

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node acts as a placeholder for all the BDS-specific information regarding BCMCS. BCAST Terminals MAY support this node and its sub-nodes.

F.3.38 <X>/BDSEntryPoint/<X>/BCMCS/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a list of Networks recognized by the terminal.

F.3.39 <X>/BDSEntryPoint/<X>/BCMCS/<X>/NetworkID

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains network region identification information for BCMCS. The Value of this node is a string comprising the concatenation of 1 (in the case of subnet) or 1, 2, or 3 (in the case of SID/NID/PZID) hexdecimal numbers, each number prefixed by characters ‘s’ (subnet), ‘S’ (SID), ‘N’ (NID), or ‘P’ (PZID), used by the terminal to determine when overhead channels indicate that the terminal is in a particular network region.

F.3.40 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ControllerIPAddress

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains the IP address of the BCMCS Controller, saving DHCP bootstrap time.

F.3.41 <X>/BDSEntryPoint/<X>/BCMCS/<X>/SGMulticastAddress

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node contains a string in the form of "<Hostname/Address>:<Port>", much like an URL without a scheme prefix. For example, the contents might be "BCMCS.example.com:83" or "129.63.44.2:8000".

F.3.42 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ROHCU

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains ROHC Unidirectional (ROHC-U) parameters for BCMCS IP multicast communication. If the node is not present, compression is not enabled.

F.3.43 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ROHCU/MaxCID

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node indicates the maximum number of CIDs used by ROHC-U.

F.3.44 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ROHCU/LargeCIDs

Status	Occurrence	Format	Min. Access Types
Required	One	bool	Get

This leaf node is true when large CIDs (1 or 2 bytes) are used, otherwise it is false and small CIDs (0 or 1) are used.

F.3.45 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ROHCU/MaxHeaderSize

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node contains the maximum header size, in octets, that can be compressed.

F.3.46 <X>/BDSEntryPoint/<X>/BCMCS/<X>/ROHCU/MRRU

Status	Occurrence	Format	Min. Access Types
Required	One	int	Get

This leaf node contains the size of the Maximum Reconstructed Reception Unit, in octets, that the decompressor is expected to reassemble from segments. Value 0 means that no segment headers are allowed on the channel.

F.3.47 <X>/BSMSelector

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	node	Get

This interior node contains information about the BSMSelector structures associated with the BSM of the Home or Roaming Broadcast Service Provider.

F.3.48 <X>/BSMSelector/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for sets of BSMSelector information.

F.3.49 <X>/BSMSelector/<X>/Name

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	chr	Get

This leaf node specifies a user readable name of BSMFilterCode associated with the BSM of the Home or Roaming Broadcast Service Provider of the user.

F.3.50 <X>/BSMSelector/<X>/BSMFilterCode

Status	Occurrence	Format	Min. Access Types
Required	One	xml	Get

This leaf node specifies the value of the

BSMFilterCode associated with the BSM. This value is used in comparisons against the BSMFilterCode values in BSMSelectors in Service Guide Delivery Descriptors and RoamingRules. The value is a BSMFilterCode XML structure as defined in [BCAST10-SG] section 5.4.1.5.2.

F.3.51 <X>/BSMSelector/<X>/IsHomeBSM

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

This leaf node specifies whether the BSM that is associated with the BSMSelector belongs to the Home Broadcast Service Provider of the user. Absence means “false”.

F.3.52 <X>/BSMSelector/<X>/HomeServiceProvisioningRequestAddress

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	chr	Get

This leaf node specifies the address (URL) of the BSM the terminal can use to issue Service Provisioning requests, as defined in section 5.1 of the present document. This address is used when the leaf node "<X>/Roaming/<X>/UseVisitedServiceProvisioningMode" is set to "false". This leaf node SHALL be present in case the leaf node <X>/BSMSelector/<X>/IsHomeBSM is set to "true".

F.3.53 <X>/BSMSelector/<X>/RoamingRules

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node contains a list of RoamingRule structures associated with the BSMSelector.

F.3.54 <X>/BSMSelector/<X>/RoamingRules/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node acts as a placeholder for a list of RoamingRule structures associated with the BSMSelector,

F.3.55 <X>/BSMSelector/<X>/RoamingRules/<X>/Rule

Status	Occurrence	Format	Min. Access Types
Required	One	xml	Get

This leaf node that contains a RoamingRule

,given as a RoamingRule XML structure as defined in section 5.4.1.5.2 of [BCAST10-SG]. This element enables the use of OMA DM as a method to manage and update roaming rules at the terminal. This leaf node SHALL apply for <X>/BSMSelector elements which have <X>/BSMSelector/IsHomeBSM set to "false".

F.3.56 <X>/Roaming

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

This interior node is a container for Roaming structures.

F.3.57 <X>/Roaming/<X>

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrMore	node	Get

This interior node is a placeholder for a list of Roaming structures.

F.3.58 <X>/Roaming/<X>/HomeRoamingRuleRequestAddress

Status	Occurrence	Format	Min. Access Types
Required	One	chr	Get

This leaf node specifies the URL of the default Server to which the terminal can send RoamingRule Requests related to the BSMSelector in case no other contact points are signalled in the Service Guide Delivery Descriptors associated with BSMSelector, or, in case the <X>/Roaming/<X>/ForceHomeRoamingRuleRequestAddress is set to “true”.

F.3.59 <X>/Roaming/<X>/ForceHomeRoamingRuleRequestAddress

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

This leaf node specifies whether the Terminal SHALL override any other RoamingRuleRequestAddresses and always contact the address represented by <X>/Roaming/<X>/HomeRoamingRuleRequestAddress for Roaming Requests.

In case its value is “true”, theTerminal SHALL always use <X>/Roaming/<X>HomeRoamingRuleRequestAddress when sending a RoamingRule Request message. In case its value is “false”, the Terminal uses <X>/Roaming/<X>/HomeRoaming-RuleRequestAddress as the backup address in case the BSMSelector in SGDD does provide any other addresses for RoamingRule Requests. In the absence of this node, default value “true” is assumed.

F.3.60 <X>/Roaming/<X>/IgnoreUnIdentifiedBSM

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

This leaf node specifies whether Terminal SHALL ignore fragments that are not associated with BSMSelector(s).

If its value is “true”, the Terminal SHALL ignore fragments that are not associated with any BSMSelector. If its value is “false”, the Terminal MAY interpret, handle, access and render fragments that are not associated with any BSMSelector without any restrictions. In the absence of this, default value “true” is assumed if the terminal has any nodes of “<X>/BSMFilterCode” present. Otherwise default value “false” is assumed.

F.3.61 <X>/Roaming/<X>/UseVisitedServiceProvisioningMode

Status	Occurrence	Format	Min. Access Types
Required	ZeroOrOne	bool	Get

This leaf node specifies whether Terminal SHALL initiate the service provisioning requests through Visited BSM or Home BSM.

In case its value is “true”, the Terminal SHALL initiate the service provisioning requests through the Visited BSM. If its value is“false”, the Terminal SHALL initiate the service provisioning requests through the Home BSM. Default value “true” is assumed.

F.3.62 <X>/Ext

Status	Occurrence	Format	Min. Access Types
Optional	ZeroOrOne	node	Get

Inside this interior node, vendor specific information related to BCAST can be placed (vendor meaning application vendor, device vendor, OS vendor etc.). Usually the vendor extension is identified by vendor specific name under the ’Ext’ node. The tree structure under the vendor extension is not defined and can therefore include a non-standard sub-tree.

Appendix G. Guidelines for extending the XML schemas in future versions of BCAST

This appendix describes the extension rules which MUST be obeyed to ensure that the XML schemas defined in future versions of BCAST keep backward compatibility.

Future versions of BCAST SHALL make sure that extensions are defined in a backward compatible way such that decoders which are not aware of these extensions can safely ignore them but still are provided all expected information. To ensure this, the following rules SHALL be obeyed when extending a BCAST XML schema in future versions of BCAST:

1) Derivation-by-extension MAY be used to derive new types from existing ones, in accordance with the rules set out in [XMLSchema].
2) Wherever possible, an extended schema SHALL only add functionality and not replace existing functionality. This will allow a decoder which is only aware of a previous version to maximally understand an instance of the extended version.
3) Existing element names SHALL never be re-used for new elements. New element names SHALL be defined under their own XML namespace.
4) Extended versions of a BCAST XML schema SHALL use a namespace identifier with a different <version> indicator but with the same <prefix>.

If a desired extension can not be done in accordance with the above rules, it is REQUIRED not to extend existing elements or types but to define new ones or to specify new signalling such that decoders which do not support the extension are able to ignore them.

Appendix H. Media-Type Registrations

H.1 Media-Type Registration Request for application/vnd.oma.bcast.sprov+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.sprov+xml
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:
Service Provisioning messages are passive, meaning they do not contain executable or active content which may represent a security threat. The format does not include confidential fields. As Service Provisioning messages convey information which services a user accesses, there is some risk that unintentional information may be exposed. 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 Provisioning 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 – Mobile Broadcast Services, especially section 5.1. Available from http://www.openmobilealliance.org

Applications, which use this media type:
OMA BCAST Services

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

Intended usage: Limited use.
Only for usage with Service Provisioning for Mobile Broadcast Services, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

Intended usage: Limited use.
Only for usage with Service Provisioning for Mobile Broadcast Services, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

H.2 Media-Type Registration Request for application/vnd.oma.bcast.drm-trigger+xml

This MIME type registration is obsolete as the DRM Trigger definition has been removed from BCAST 1.0 specifications.

H.3 Media-Type Registration Request for application/vnd.oma.bcast.smartcard-trigger+xml

This MIME type registration is obsolete as the Smartcard Trigger definition has been removed from BCAST 1.0 specifications.

H.4 Media-Type Registration Request for application/vnd.oma.bcast.imd+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.
Type name: application
Subtype name: vnd.oma.bcast.imd+xml
Required parameters: none
Optional parameters: none
Encoding considerations: binary

Security considerations:
InteractivityMediaDocument data are active, meaning that upon the reception of the InteractivityMediaDocument, the terminal will interpret it and act based on the commands and structures in the document. There is a possibility that a maliciously formed InteractivityMediaDocument will cause unwanted operations. To protect the user and terminal against these operations, the terminal should notify or prompt the user in case the interpretation of InteractivityMediaDocument will cause a critical operation at the terminal (sending outbound data, accessing system areas, etc.). InteractivityMediaDocument data do not include confidential fields. 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 interactivity 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 – Mobile Broadcast Services, especially section 5.3.6.1. Available from http://www.openmobilealliance.org

Applications, which use this media type:
OMA BCAST Services

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

Intended usage: Limited use.
Only for usage with Service Interactivity for Mobile Broadcast Services, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

H.5 Media-Type Registration Request for application/vnd.oma.bcast.notification+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.notification+xml
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:
BCAST Notification message data are passive, meaning they do not contain executable or active content which may represent a security threat. The format does not include confidential fields. 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 notification 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 – Mobile Broadcast Services, especially section 5.14. Available from http://www.openmobilealliance.org

Applications, which use this media type:
OMA BCAST Notification client

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

Intended usage: Limited use.
Only for usage with the BCAST Notification message, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

H.6 Media-Type Registration Request for application/vnd.oma.bcast.provisioningtrigger

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.provisioningtrigger

Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:

BCAST Provisioning Trigger message data are passive, meaning they do not contain executable or active content which may represent a security threat. The format does not include confidential fields. 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 trigger messages 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 – Mobile Broadcast Services, especially section 5.1.8. Available from http://www.openmobilealliance.org

Applications, which use this media type:

OMA BCAST Service Provisioning Client

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

Intended usage: Limited use.

Only for usage with the BCAST Provisioning Trigger message, which meets the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

H.7 Media-Type Registration Request for application/vnd.oma.bcast.roaming+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	vnd.oma.bcast.roaming+xml
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:

BCAST Roaming message data are passive, meaning they do not contain executable or active content which may represent a security threat. The format does not include confidential fields. 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 roaming information within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – Mobile Broadcast Services, especially section 5.7.1. Available from http://www.openmobilealliance.org

Applications, which use this media type:

OMA BCAST Roaming client

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

Intended usage: Limited use.

Only for usage with the BCAST Roaming message, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org