ITUPublications

International Telecommunication Union
Radiocommunication Sector

# Report ITU-R BT.2267-13

**(10/2026)**

BT Series: Broadcasting service (television)

# Integrated broadcast-broadband systems

## Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

## Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from https://www.itu.int/ITU-R/go/patents/en where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

### Series of ITU-R Reports

(Also available online at https://www.itu.int/publ/R-REP/en)

| Series | Title |
|---|---|
| **BO** | Satellite delivery |
| **BR** | Recording for production, archival and play-out; film for television |
| **BS** | Broadcasting service (sound) |
| **BT** | **Broadcasting service (television)** |
| **F** | Fixed service |
| **M** | Mobile, radiodetermination, amateur and related satellite services |
| **P** | Radio-wave propagation |
| **RA** | Radio astronomy |
| **RS** | Remote sensing systems |
| **S** | Fixed-satellite service |
| **SA** | Space applications and meteorology |
| **SF** | Frequency sharing and coordination between fixed-satellite and fixed service systems |
| **SM** | Spectrum management |
| **TF** | Time signals and frequency standards emissions |

***Note**: This ITU-R Report was approved in English by the Study Group under the procedure detailed in Resolution ITU-R 1.*

*Electronic Publication*
Geneva, 2026

© ITU 2026

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without written permission of ITU.

# REPORT ITU-R BT.2267-13*

## Integrated broadcast-broadband systems[1, 2]

(05/2013-11/2013-04/2014-11/2014-02/2015-07/2015-2016-2017-2018-04/2019-07/2019-03/2021-2023-2026)

## 1 Introduction

In a media scenario where convergent TV receivers are able to handle not only the broadcast signal but also applications delivered by broadband IP telecommunication services, there are opportunities to drive user engagement and to maximize the end-user's satisfaction by offering a range of new services. A system which enables to offer such services is called Integrated broadcast-broadband (IBB) system.

These IBB services should be able to extend the traditional broadcasting using any telecommunication mechanism available in order to bring new, high-quality, interactive and complementary content to the end-user. Such systems should be open, customizable and easy to use, while maintaining the traditional TV user experience, copyright and the broadcaster audiovisual integrity. The major point of difference between IBB systems and the web-based services is the capability to combine multi-functional IBB applications with traditional broadcast programmes or services.

This Report is intended to provide the information on IBB systems which enable to offer IBB services and considerations to build IBB services. This Report consists of three parts. In Part 1, an overview of IBB systems is described. Each Annex in Part 1 describes specific information on each IBB system. In Part 2, general considerations to build IBB services are described. Part 2 focuses on things to be considered to build IBB services on the IBB systems which are designed to employ benefits of both broadcast and broadband as much as possible. Part 3 assists the translation of applications among IBB systems in order to allow the exchange of applications covered by the IBB systems considered in this Report and to deploy these applications in an area where other systems are used. Taking this combination further, Report ITU-R BT.2568 describes application-oriented television broadcasting, one possible evolution of IBB systems in which applications, rather than the traditional selection of channels, become the viewer's entry point to broadcast content, enabling personalization from the beginning of the viewer's journey.

Part 1: Overview of IBB systems

Annex 1 – The hybrid broadcast broadband television (HbbTV) system

Annex 2 – The hybridcast system

Annex 3 – Integrated broadcast-broadband system based on enhancement of data broadcasting

Annex 4 – Integrated broadcast-broadband system based on the Ginga middleware

Annex 5 – TV Open Platform for Smart media (TOPSmedia) system

---

* This Report should be brought to the attention of ITU-T Study Group 21.

[1] Integrated broadcast-broadband system is a system in which broadcasting operates in parallel with broadband telecommunication systems and provides an integrated experience of broadcasting and interactivity by combining media content, data and applications from sources authorized by the broadcaster.

[2] This Report is a part of the family of Recommendations and Reports on IBB systems.

Part 2: Considerations to build IBB services
1 Consideration on use of Integrated broadcast-broadband technologies on mobile devices
2 On-demand content playback
Part 3: Harmonization of applications among different IBB systems
1 Introduction
2 Approaches for harmonization
3 Comparison of application environment of various IBB systems
4 Harmonization of second screen handling

## 2 Abbreviations

AAC Advanced audio coding
ABNT *Associação Brasileira de Normas Técnicas* (Brazilian technical standards association)
AIT Application information table
ANSI American National Standards Institute
API Application programming interface
ARIB Association of radio industries and businesses
AVC Advanced video coding
BML Broadcast markup language
CATV Cable television
CC Common core
CE Consumer electronics
CEA Consumer Electronics Association[3]
CENC Common encryption
CI Common interface
CPU Central processing unit
CSS Cascading style sheets
DASH Dynamic adaptive streaming over HTTP
DIAL Discovery and launch
DMIPS Dhrystone millions of instructions per second
DOM Document object model
DRM Digital rights management
DSL Digital subscriber line
DSM-CC Digital storage media command and control
DTV Digital television

[3] In 2015, CEA was renamed CTA (Consumer Technology Association).

| | |
|---|---|
| DVB | Digital video broadcasting |
| EBU | European broadcasting union |
| EIT | Event information table |
| EPG | Electronic program guide |
| ETSI | European telecommunications standards institute |
| FTTH | Fiber-to-the-home |
| GIF | Graphics interchange format |
| GPU | Graphics processing unit |
| HDR | High dynamic range |
| HDTV | High-definition television |
| HE-AAC | High-efficiency advanced audio coding |
| HEVC | High-efficiency video coding |
| HFR | High frame rate |
| HLS | HTTP live streaming |
| HTML | Hypertext markup language |
| HTTP | Hypertext transfer protocol |
| HTTPS | Hypertext transfer protocol secure |
| IBB | Integrated broadcast-broadband |
| IDL | Interface description language |
| IMSC | Internet media subtitles and captions |
| IP | Internet protocol |
| IPTV | Internet protocol television |
| IPTVFJ | IPTV forum Japan |
| ISDB | Integrated services digital broadcasting |
| ISOBMFF | ISO base media file format |
| JPEG | Joint photographic experts group |
| JSON | JavaScript object notation |
| LTE | Long-term evolution |
| MHP | Multimedia home platform |
| MMT | MPEG media transport |
| MNG | Multiple-image network graphics |
| MPD | Media presentation description |
| MPEG | Motion picture expert group |
| MSE | Media source extensions |
| NCL | Nested context language |
| NGA | Next generation audio |

| | |
|---|---|
| OFF | Open font format |
| OIPF | Open IPTV forum |
| OTT | Over the top |
| PMT | Program map table |
| PNG | Portable network graphics |
| PVR | Personal video recorder |
| REST | Representational state transfer |
| ROM | Read-only memory |
| RTSP | Real time streaming protocol |
| SDR | Standard dynamic range |
| SI | Service Information |
| SNS | Social networking service |
| TCP | Transmission control protocol |
| TS | Transport stream |
| TT | Timed text |
| TTA | Telecommunications technology association |
| TTML | Timed text markup language |
| UDP | User datagram protocol |
| UHDTV | Ultra-high-definition television |
| UI | User interface |
| UMTS | Universal mobile telecommunications system |
| UPnP | Universal plug and play |
| URI | Uniform resource identifier |
| URL | Uniform resource locator |
| UTF | Unicode transformation format |
| VOD | Video on demand |
| W3C | World Wide Web Consortium |
| WebVTT | Web video text tracks |
| WLAN | Wireless local area network |
| WOFF | Web open font format |
| XHTML | Extensible hypertext markup language |
| XML | Extensible markup language |

PART 1

# Overview of integrated broadcast-broadband systems

TABLE OF CONTENTS

*Page*

Annex 1 – The hybrid broadcast-broadband television (HbbTV) system ..... 8

1 Introduction ..... 8

2 Combining broadcasting and Internet services ..... 8

3 The usage of HbbTV by network operators as well as portals and platforms of a diversity of stakeholders ..... 9

4 Some outlook on the standardization of HbbTV and its actual market position ..... 9

Attachment to Annex 1 – HbbTV – Technical information ..... 12

1 Conceptual considerations and requirements ..... 12

2 Technical concept of HbbTV ..... 13

3 Developing and testing HbbTV ..... 17

4 Extended features in Version 1.5 of the HbbTV specification (HbbTV 1.5) ..... 18

5 Version 2.0 of the HbbTV specification (HbbTV 2.0) ..... 18

6 Applying HbbTV to non-DVB broadcasting systems ..... 19

Annex 2 – The Hybridcast system ..... 20

1 Introduction ..... 20

2 Specification structure ..... 20

3 System model ..... 20

4 Application model ..... 22

4.1 Application type ..... 22

4.2 Application boundary and permission ..... 22

5 Application control signal ..... 23

5.1 Application Information Table (AIT) ..... 23

5.2 Coexistence with data broadcasting ..... 23

6 Hybridcast receiver ..... 24

7 HTML5 environment and additional APIs ..... 25

8 Companion device collaboration ..... 26

*Page*

9 Advanced VOD Playback ..... 27

10 Non broadcast-oriented managed application ..... 28

11 Broadcast-independent managed application ..... 29

12 Accurate synchronized presentation ..... 29

13 UHDTV support ..... 29

14 Service examples ..... 32

14.1 Multilingual closed caption ..... 32

14.2 Social TV ..... 32

14.3 Language study ..... 33

14.4 Multilingual caption with companion devices ..... 34

14.5 Non broadcast-oriented managed application ..... 36

14.6 Interactive UHDTV ..... 37

14.7 First Hybridcast services ..... 39

14.8 Other services offered ..... 42

15 Use cases of companion devices ..... 42

15.1 Interactive manipulation with broadcast programme ..... 42

15.2 Interactive presentation of related information to broadcast programme ..... 43

15.3 Connection with web sites ..... 43

15.4 Synchronized multi-screen presentation of video streams ..... 43

15.5 Common companion application ..... 44

15.6 Common companion application to execute an application on the receiver ..... 44

16 Multi-resolution video services using Hybridcast ..... 47

16.1 MPEG-DASH player for Hybridcast ..... 47

16.2 Use cases of Multi-resolution video services using Hybridcast ..... 47

17 Broadcast-based personalized video services using Hybridcast ..... 49

17.1 Background ..... 49

17.2 Use case ..... 49

18 Object-based sound services using Hybridcast ..... 52

19 Receiver system using broadcast-oriented managed applications and broadcast-independent managed applications ..... 53

*Page*

Annex 3 – Integrated broadcast-broadband system based on enhancement of data broadcasting ..... 54
1 Introduction ..... 54
2 Internet-based interactive data broadcasting ..... 55
3 Switching to IPTV/Internet-TV services ..... 56
4 Direct access to IPTV/Internet-TV services ..... 56
5 Compatibility consideration ..... 58
6 Future development ..... 58
Annex 4 – Integrated broadcast-broadband system based on the Ginga middleware ..... 58
1 Introduction ..... 58
2 Ginga position ..... 59
3 Ginga architecture ..... 63
4 Integrated broadcast and broadband delivery ..... 64
5 Secondary devices (second screens) ..... 65
6 Application control ..... 68
7 Developing and testing Ginga and applications ..... 69
Annex 5 – TV Open Platform for Smart media (TOPSmedia) system ..... 70
1 Introduction ..... 70
2 Application Model and Lifecycle ..... 72
3 W3C Profile for Smart TV ..... 74
4 Extended APIs ..... 74
5 Application Signalling and Packaging ..... 75
6 Other features ..... 76
7 Service examples ..... 76
8 Future works ..... 78
Attachment to Annex 5 – Cloud TV Open Platform for Smart media (COPSmedia) system ..... 79
1 Introduction ..... 79
2 TOPSmedia profile ..... 80
3 Cloud service transmission ..... 80
4 Interoperability protocol ..... 82
5 Service example ..... 83

Integrated Broadcast-Broadband (IBB) systems combine both broadcast and broadband technologies, maximizing benefits of combination, to provide high quality, flexible, interactive, and personalized services. Broadcast provides efficient delivery of high-quality information to a number of viewers at the same time. Broadband enables flexible and personalized services such as Social Network Services (SNS). Although it has been long time to use telecommunication network for interactive broadcast services, recent development of broadband related technologies enables much more than sending simple messages from a receiver to a broadcaster. Recent advances of broadband related technologies allow to employ high speed network, powerful server-side processing and inter-devices communication, which brings modern and rich web-based services into broadcast environment. With these technologies, requirements of IBB systems are defined in Recommendations ITU-R BT.2037 and ITU-R BT.2053.

In Part 1 of this Report, information on various IBB systems is provided in each Annex. Each Annex describes how each IBB system combines these technologies to offer IBB services.

# Annex 1

# The hybrid broadcast-broadband television (HbbTV) system

## 1 Introduction

In connected TV, the consumer has access to all IP services via a standard IP interface. In the home, the IP connection is typically accomplished via WLAN or Ethernet (fed over DSL or a specific Cable TV Modem). In case of portable or mobile TV reception, the Internet connection may be provided by a UMTS or LTE modem built into a portable/mobile TV-set. HbbTV as a more broadcast-centric type of connected TV allows seamless interlinking of broadcast and IP services making use of the architecture of connected TV. HbbTV also supports services not linked to broadcasting, thus making it applicable for platforms, portals, and to what is known today as “OTT-services”.

## 2 Combining broadcasting and Internet services

A Connected (or Smart) TV set is not necessarily apt for truly interactive viewing experiences. While, in general, all Connected TV sets have two inputs, one for the broadcast signal (TV tuner) and one for the Internet (Ethernet/WLAN) connection, they do not necessarily offer converged services by making use of both distribution paths. Such a device is usually only equipped for access to proprietary portals for content and applications via the Internet. In addition, some include a slimmed-down browser for viewing regular Web pages, however, usually with limited functionality (which commonly leads to odd user experiences). This cannot be considered as converged equipment; it is merely a multi-purpose device that JUST allows the viewing of broadcast television content OR using separated and limited add-on functionalities through the Internet connection on the same screen.

For the true IBB services enabling a seamless user experience, an “engine” is required that links the broadcast content offered via satellite, terrestrial or CATV networks and the Internet content offered via the interaction channel, be it via Ethernet on DSL or via Ethernet on CATV or via mobile broadband networks such as LTE – or via any other IP connection. HbbTV provides such an engine. HbbTV encompasses the necessary signalling and includes a CE-based browser that has combined access to both, the data in the broadcast stream as well as to the services, applications and content provided via the Internet.

For a Connected TV set that is equipped with the HbbTV function, the consumer just has to push the red button on the TV's remote control to make the HbbTV launch page of the corresponding broadcaster visible. Subsequently, the end-user can select all services (incl. video-on-demand and search functions) that are offered by or via this broadcast service specific portal. Example: A user would like to have more information on, say "Napoleon". The result of the search will be a list of all Napoleon-related video clips that are stored and offered by the collaborating broadcasters. Potentially, sound radio programmes and adapted Web-pages (incl. pictures and text files) could also be included in the result list. Viewing of the retrieved content is currently accomplished on the TV set but may in future also happen on a second screen, e.g. on a tablet computer (ref. also text on HbbTV 2.0 in § 4 below).

## 3 The usage of HbbTV by network operators as well as portals and platforms of a diversity of stakeholders

HbbTV is a business-model neutral technology platform. Thus, HbbTV can be used also to operate and provide service portals by a network operator. Some smaller cable TV operators in Switzerland have deployed HbbTV-based portals offering a broad variety of services on any HbbTV-equipped device. The service channel may be used to signal an application which, by the HbbTV auto-launch functionality, is immediately kicked-off when the device tunes to the service channel. An example is M7 Deutschland's "Kabelkiosk choice", an HbbTV-based service portal that is pre-configured for cable network operators.

Generally, HbbTV is, in this market segment, used to build a platform for the variety of small and local TV broadcasters. Some of them may not have the financial resources to pay for a 24/7 broadcast link, but share it with others or even offer Internet-streaming only. The HbbTV-based portal allows for providing a harmonized view to all offered services and automatically redirects the consumer to the selected service regardless of whether it is delivered via broadcast or via the Internet.

Finally, HbbTV is not limited to broadcast centric services. It can likewise be used to build application portals and platforms which provide Internet-only services. Stakeholders are manufacturers but also independent application portals. It is also suitable for specific OTT-services, which are not necessarily accessed via a portal, but directly through predefined buttons on a remote control ("Netflix-button"). Currently, the great majority of HbbTV services are broadcast-centric and thus link a given TV programme with the portal of the programme provider.

## 4 Some outlook on the standardization of HbbTV and its actual market position

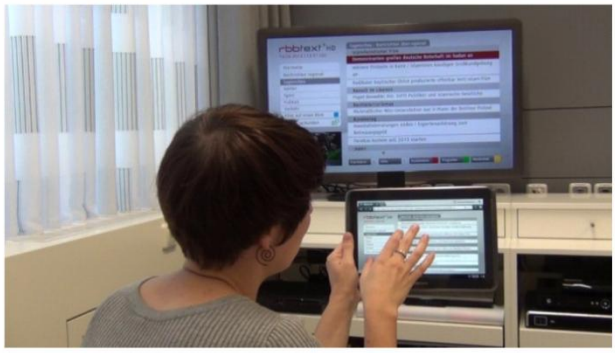
HbbTV was developed in 2009 and first standardized by ETSI in 2010. Version 1.5 of the HbbTV specification has been published by the HbbTV Consortium in April 2012. Standardization of HbbTV 1.5 as ETSI TS 102 796 v1.2.1 took place by ETSI in November 2012. Amongst other new features, adaptive streaming (in line with MPEG-DASH) is supported (see § 4 of the Attachment to Annex 1 for more details).

In 2015 a functionally significantly extended version of the HbbTV standard has been published as HbbTV 2.0. Meanwhile this version has been updated to version HbbTV 2.0.2, details of which are described in § 5 of the Attachment to this Annex 1. Amongst others, the European project HBB-NEXT (which is co-funded by the European Commission, ref. www.hbb-next.eu) has been supporting the technical specification work for HbbTV 2.0. The HBB-NEXT work includes links to social networks like Facebook and provides for enhanced multimodal user interfaces (e.g. speaker identification, voice control, etc.). HBB-NEXT also looks at new recommendation features, both for groups and individuals. Especially, the efforts in the area of synchronization of broadcast and Internet-delivered content will be helpful to all but will be decisive for people with disabilities. Synchronization will allow for the time-correct display of additional services such as the video

representation of a sign language interpreter[4]. Synchronization will also help to overcome the problem of delayed subtitles (as occurring today when subtitles create life captions, e.g. by "re-speaking"). Improved accessibility for people with disabilities is also reached through yet another strand of HBB-NEXT which looks at second screen services and usage (log-in of the handset is via the main TV set).

Here, for example, a user might listen to a synchronized audio description with a second device (via earphones) while the other users in the room listen to the original sound on the TV set (or vice versa). The technical solution of connecting the second screen was predominantly developed within the EU project FI-CONTENT (co-funded by the European Commission, ref. http://www.fi-content.eu/).

FIGURE 1-1



*The second screen solution allows the usage of an HbbTV application on the main screen, on the second screen, e.g. a tablet PC or smartphone, or on both screens (as it is the case in this photograph).*




*Menu to select size, appearance and position of HbbTV delivered subtitles (Source: HBB-NEXT)*

Report BT.2267-1-01

In principle, HbbTV can be used to provide any access service required: signer video, audio description, spoken subtitles, multi-lingual text subtitles, multi-language sound tracks or additional sound tracks with clear(er) audio dialogues, etc. Today, HbbTV is already of help to people with viewing difficulties as the new text-services provided by HbbTV are much better to read than the conventional videotext service and, in some instances, already offer personalization options for further improving readability through extended font sizes and various colour options (adapting colour contrast to individual needs). Further access services like, for example, an application which allows for individual configuring of subtitles in terms of their size, position or background are in experimental or pre-operational stage.

[4] Defined in Recommendation ITU-R BT.2053 as "An application that is part of the integrated broadcast and broadband (IBB) DTV service tuned to by the user at a given time."

FIGURE 1-2




*The inscription says: Via Internet, a signer video is available for this programme - switch it on now?*




*Example of a signer video (delivered via the Internet) superimposed with the main broadcast video*

Report BT.2267-1-02

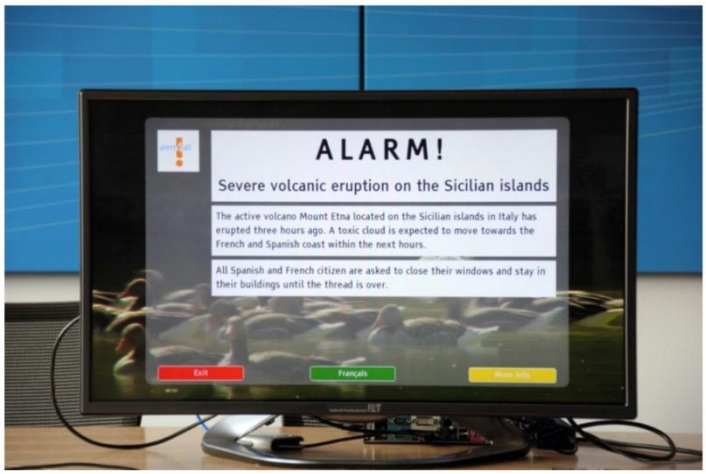
But foremost, HbbTV is used for information, education and entertainment (e.g. catch-up TV). It is also used for commercial applications (music download, online shopping, (targeted) advertisement, etc.).

It was also demonstrated that HbbTV represents a prime means to alert the general public in case of a crisis. The EU project Alert4All (http://www.alert4all.eu/) has developed an according protocol for a web-service available to broadcasters. The alert information can be introduced in the data carousel (DSM-CC) within the transport stream of the digital TV signal and triggers an HbbTV auto-launch function.

No Internet connection is required. The messages can be provided in various languages selectable by the end-user. If an Internet connection is available, supporting information such as detailed maps or additional instructions, which are not mandatory to handling the crisis, is offered to the viewers for retrieval via the IP path [7]. The colour fields indicated correspond to the colour buttons on the remote control. Figure 1-3 depicts a test example of automatic popup of an HbbTV alert message (HbbTV test application of the Alert4All project).

FIGURE 1-3

**A test alert displayed on a common TV set that is capable of HbbTV**




Report BT.2267-1-03

The alert information remains in the DSM-CC as long as necessary. The application is thus running automatically once a TV service is chosen by a consumer. It is only if an alert is sent with a "stream event" that the HbbTV application will automatically be displayed on the TV screen. By default, the alert is displayed in the same language as the ongoing TV service. Other languages (as provided by the broadcaster) are accessed via (in this example green) button.

The data composing the essential information from the basic Alert4All HbbTV application has been purposely kept small – in the order of 100 KB per alert. The data required for disseminating such alerts in less than six seconds is in the order of only 100 kbit/s. This is a small overhead for an HDTV channel that is typically transmitted on average at 8 Mbit/s.

Up to now, a high number of applications have been developed for HbbTV (from pre-views and catch-up TV to additional access services and commercial applications). For example, more than 50 applications are known in Germany. A detailed description of HbbTV is provided in the Attachment to this Annex 1.

Currently, consumer equipment manufacturers offer a large number of models of HbbTV-enabled TV sets and set-top boxes.

HbbTV has gained a large momentum in European markets. Broadcasters in fourteen European countries have already started to offer HbbTV applications on a regular basis or have announced to start soon. With Australia launching regular HbbTV services in 2014, HbbTV sees a major deployment outside Europe. Worldwide interest is growing as HbbTV can easily be adapted to any digital television system. Interest ranges include those from China, Asia/Pacific including Japan to the United States of America and South America.

# Attachment to Annex 1

# HbbTV – Technical information

## 1 Conceptual considerations and requirements

For a broadcast service provider entering the IBB era three main problems occurred with first generation types of so-called "hybrid TV" sets (that allowed for some kind of Internet access):

1) each manufacturer used its own browser profile and its own set of other technical features like streaming protocols or remote control key sets. This means that the services had to be re-authored for each set manufacturer individually, and this requires not only an operational effort but also a conceptual one if, for example, graphic options or key sets differ. This kind of technical fragmentation was a considerable obstacle for a dynamic development on the content side;

2) the entry portal to the Internet applications is typically controlled by the manufacturer and no one but the manufacturer was able to add or replace applications on that portal. Consumers were locked in a kind of a walled garden;

3) although using the same screen, both worlds – the broadcasting and the Internet world – remained separated in the end. Either there is a hard switching between both modes – when watching TV the connection to the Internet is lost and when the browser is active the TV signal is lost – or the TV picture is overlaid by unrelated Web content. A combination of

service components from both networks was not foreseen thus losing the potential for true IBB services.

Therefore, at the beginning of 2009 a number of market partners joined forces within a pan-European project to develop a technical scenario that allows to address these issues and to provide a standardized and functional framework for hybrid television. The targets for this development were:

- to provide an open and standardized HTML-based system in order to allow efficient content development by leveraging existing on-line services independent from specific manufacturers or platform operators;
- to use, as much as possible, existing standardized components in order to gain acceptance and time-to-market benefits;
- to specify only a minimum set of features required for all basic needs, which allows an easy integration in existing hardware platforms and acceptance throughout the entire value chain;
- to allow the combination of all broadcast distribution networks with all Internet access technologies;
- to allow the creation of IBB services using broadcast services and additional resources from the Internet at the same time;
- to provide the potential to the successor of the Teletext system;
- to avoid the “conquering” of TV signals by unauthorized third party Web services;
- to be also applicable to radio services.

With a system like this, “red button” services linked to TV programmes can be implemented. Teletext-type services can be enhanced graphically in a form more adequate to the HD era. Additional services like news tickers can be provided in a much more functional and flexible way. HTML-overlays over full screen TV pictures are possible as well as the integration of a scaled TV image into a full screen application.

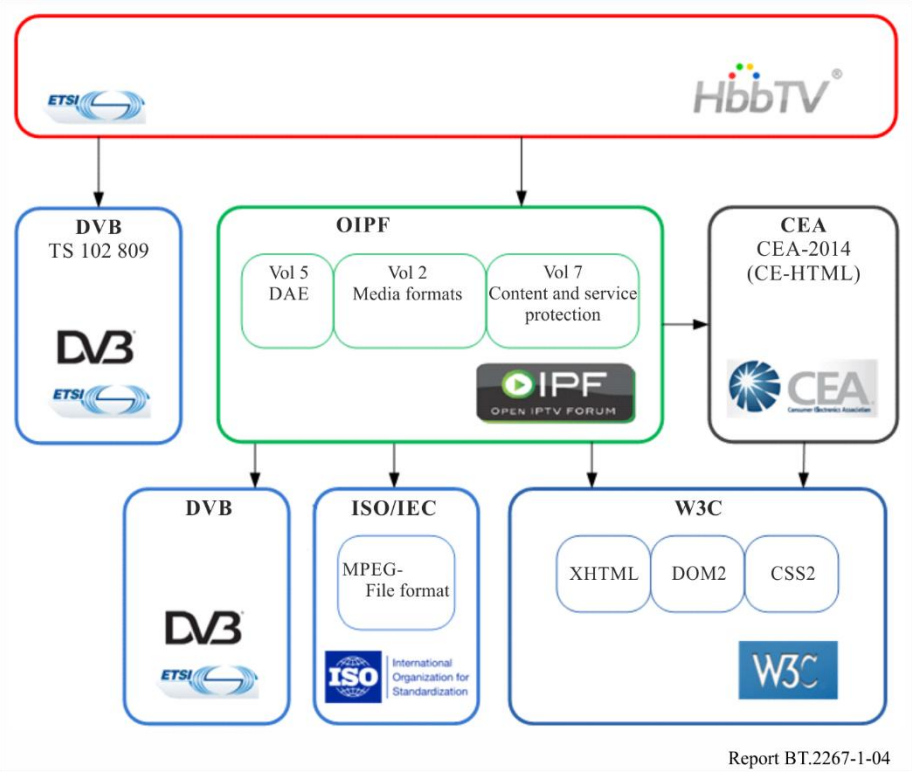
This work resulted in the definition of the HbbTV specification and in the creation of the HbbTV Consortium (ref. www.hbbtv.org).

## 2 Technical concept of HbbTV

The basic concept of the HbbTV specification, the first version of which was published by ETSI as TS 102 796 in June 2010 [1], is shown in Fig. 1-4. The specification introduces only few new technical components, but is mainly based on existing standards. In this respect, it represents more a specific profile of available technologies than a completely new technical development. Such approach is extremely valuable in terms of development cost and specifically for a short time-to-market period.

FIGURE 1-4

**Standards involved in the HbbTV concept for version 1.0**
**(Later versions include extensions to this concept, see §§ 4 and 5)**




In detail, HbbTV is based on the following three standards:

CE-HTML [2] defines the core browser functionality for HbbTV. CE-HTML is based on the common W3C Web standards and it specifies a HTML profile for CE devices. It uses XHTML 1.0, DOM 2, CSS TV profile 1.0 as well as ECMAScript-262 ("JavaScript") and is optimized for rendering HTML/JavaScript Web pages on CE devices, specifically on TV screens.

As the XMLhttpRequest object is supported, application developers are able to design HTML applications which are very similar to up-to-date Web 2.0 services. Compatibility in this respect allows to apply existing know-how, technology and experience used for standard Web development today also to HbbTV in a seamless way. So, the effort for TV integration can be kept on the level of necessary minimum of optimization for the different viewing situations.

CE-HTML also contains elements like the definition of the key codes for the usual TV remote control. On the contrary, CE-HTML does not convey any interfaces to the "DVB world". These are provided by the browser specification of the Open IPTV Forum [3] which was published in January 2009. This specification has been developed for DVB based IPTV systems, but the APIs it provides can be applied to any IBB DVB system. These APIs convey functions to combine the TV picture with HTML pages, to tune to other DVB TV or radio services, to add events to the timer list, to read DVB metadata and other DVB-related things.

The selected components from CE-HTML and the Open IPTV browser define the main functions of the HbbTV browser component.

Beyond the browser functionality, more DVB related integration capability is required. This is granted by including reference to the DVB standard "Signalling and carriage of interactive applications and services in hybrid broadcast/broadband environments", which was completed by DVB in March 2009 and subsequently published by ETSI [4]. As said by its title, this DVB standard defines the signalling of applications which shall be run in the context of specific TV or radio services

in the corresponding DVB multiplexes. In a very similar way to the Multimedia Home Platform (MHP) standard this is done via an Application Information Table (AIT) of the relevant DVB services which is pointed to from its Program Map Table (PMT). The AIT carries the signalling of all applications which are supposed to run in the context of this programme. Other applications are allowed to tune to this programme but they are stopped unless they are referenced in its AIT. Thus, it can be avoided, for example, that TV programmes are conquered by third-party applications which could do advertising or other overlays – thus misusing both the business model of the broadcaster and the confidence of the viewers in the integrity of what they see on the screen.

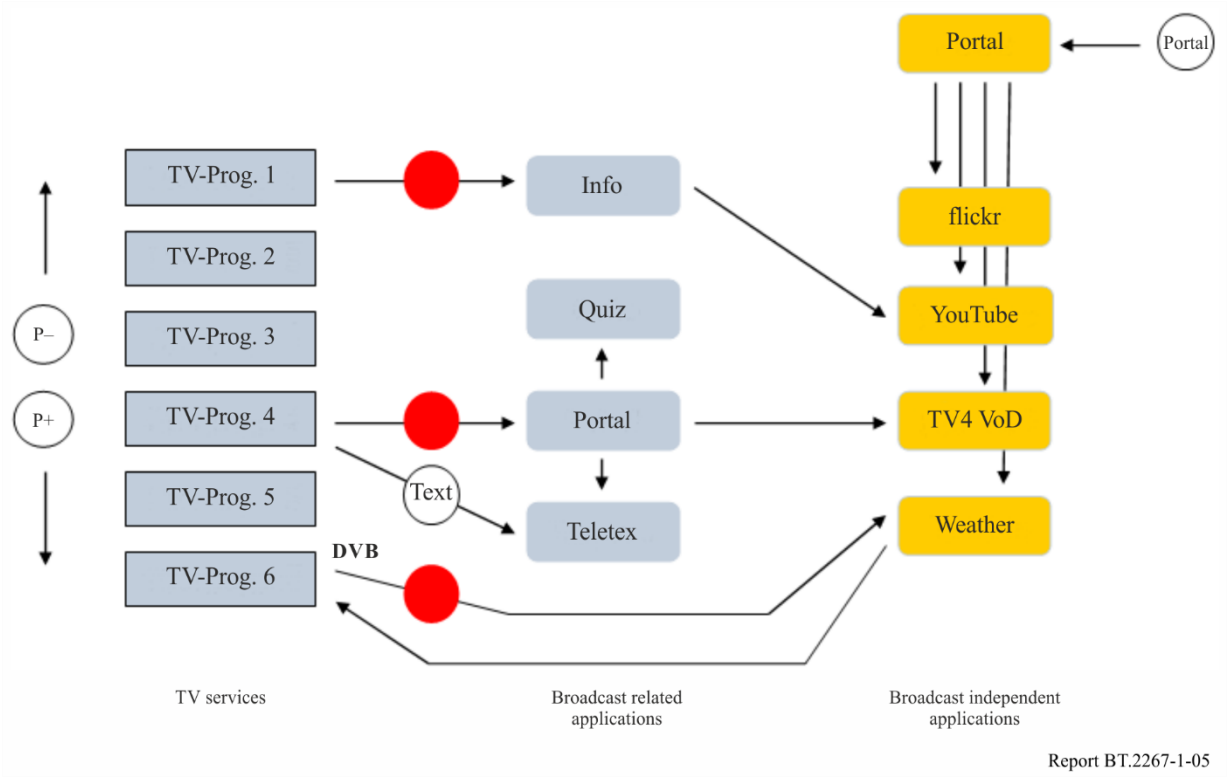
One of the applications signalled in the AIT can be marked as "autostart", which means that this application is automatically launched after tuning to the corresponding service. In order not to annoy the viewer by undesired overlays and to allow a uniform experience for application start, it has been agreed as a guideline that the application should draw a small icon including a red button first and then disappear after a few seconds. The application is then active from a technical perspective but does not draw anything on the screen until the user presses the red button.

Another signalling option for the AIT is designed to support the establishment of HbbTV as a successor of the Teletext standard. An application can be signalled as "digital Teletext application" which allows the CE manufacturer to start this specific HTML application when the familiar Teletext key is pressed on the remote control.

Figure 1-5 illustrates how HbbTV allows implementing applications which are independent of broadcasting services and how applications can be tied to broadcasting services.

FIGURE 1-5

**HbbTV applications can be tied to broadcasting services or can be independent of broadcasting services**



But HbbTV does of course not only support applications which are tied to a broadcast service ("broadcast-related applications") but also applications with no relation to any broadcast service ("broadcast-independent applications"). Such broadcast-independent applications can be TV editions of existing Web services like flickr, YouTube and very many more as they may be provided by the big brands as well as by regional service providers or even by individuals. HbbTV does not define detailed access mechanisms for broadcast-independent applications. It is left to the CE manufacturers to implement flexible portals allowing the end-users to find and access all the services they could be interested in. HbbTV portals and search functions for interesting HbbTV applications could also be provided by third-party operators. It is left open to the market development in an open retail market to find access strategies serving best the end customer's need. In vertically integrated market segments (with subsidized devices) more closed portal services may be provided.

Furthermore, ETSI TS 102 809 specifies the carriage of HbbTV applications via the DVB broadcast channel. This option is specifically interesting for HbbTV devices which are not connected to the Internet. In the market this may be the case for quite a considerable number of devices as some manufacturers plan to integrate HbbTV as a standard feature like Teletext today. So this feature may be found in devices which are not explicitly purchased for IBB use. Similar to MHP, the digital storage media command and control (DSM-CC) object carousel is used to transmit the applications. Certainly, the whole amount of data which can be distributed via broadcast channel is rather limited compared to the Internet capabilities, but may be sufficient for relatively lightweight information applications like graphically and functionally advanced Teletext services.

As a part of the DSM-CC system, "stream events" are available in HbbTV which consist of small data packets that can be transmitted synchronously to the programme signal. They allow conveying time-accurate data as, for example, questions or answers for an interactive quiz show. Timing precision of an Internet connection would be more limited for this purpose and, in addition, would put very high performance requirements on the Web servers to serve potentially millions of real time connections at the same time. Using the broadcast channel is by far more efficient for applications like this.

As a whole, the overall technical concept of HbbTV represents a pragmatic compromise which provides a flexible and universal infrastructure for a variety of broadcast-related and broadcast-independent services, but allows relatively fast integration in CE hardware which is less powerful than that of standard multimedia PCs.

The HbbTV system is designed as a full interactive system on its own. But middleware vendors can decide to integrate HbbTV into their own products which may provide more functionality like scanning routines, PVR management, EPGs and others.

The screenshots below show some examples of applications in regular operation.

FIGURE 1-6




France: France 24




Spain: Telecinco




Germany: ARD




The Netherlands: NPO

Report BT.2267-1-06

## 3 Developing and testing HbbTV

HbbTV's unique proposition in the market offers easiest application development and porting as it is based on the common set of Internet technology that is used by Internet services. Know-how related to HTML and JavaScript is widely and readily available and can easily be employed for the development of HbbTV applications. Thus, existing online services can be adapted straight-forward to the TV screen; and service providers can more or less instantaneously provide services also for the TV screen. This allows leveraging maximal synergy and effectiveness at the level of conceptual design and application development as well as at the level of integration and operation.

Technically, data delivery in an individually optimized way to PC and TV screens uses existing XML data feeds and Web servers. The uniqueness of HbbTV is to seamlessly band together linear and non-linear service elements, even personal communication means, unleashing the development of completely new hybrid services; HbbTV supports the integration of both worlds via simple API extensions as described above.

Obviously, the development of HbbTV applications does not require specific tools for all the standard application functionality. For testing the more TV-related features specifically modified PC browsers are available.

Tests allow identifying potential interoperability problems. Early-on tests accelerate the application and the device development alike. Due to the limited hardware resources of CE devices, checking application performance on consumer hardware is advisable as well. Since the first implementation phase of the HbbTV standard end of 2009, the IRT[5] has organized quarterly HbbTV interoperability workshops. Representatives from numerous companies from across the value chain – including

[5] Institut für Rundfunktechnik GmbH (R&D Establishment of the public service broadcasters in Germany, Austria and Switzerland).

broadcasters, software providers and CE device manufacturers – have since then attended these events to evaluate current HbbTV applications and implementations.

Compliance of all components to the specification will be crucial for a broad market success of the whole system. To allow a more standardized testing of HbbTV implementations, an HbbTV working group currently has developed a test suite which conveys a large set of test applications for HbbTV implementations (see http://www.hbbtv.org/pages/about_hbbtv/hbbtv_test_suite.php). This will help manufacturers to verify the compliance of their products and give confidence that a proper interoperability level can be achieved.

## 4 Extended features in Version 1.5 of the HbbTV specification (HbbTV 1.5)

Version 1.5 of the HbbTV specification [5] was adopted by the HbbTV Consortium in April 2012. "HbbTV 1.5" introduces support for HTTP adaptive streaming (based on MPEG-DASH), improving the perceived quality of video presentation on busy or slow Internet connections. It also enables content providers to protect DASH delivered content with potentially multiple DRM technologies based on the MPEG CENC specification, improving efficiency in markets where more than one DRM technology will be used. Version 1.5 significantly enhances access to broadcast TV schedule information, enabling operators to produce full 7-day electronic programme guides as HbbTV applications that can be deployed across all HbbTV receivers to provide a consistent user experience. "HbbTV 1.5" was submitted to ETSI for European standardization. Consequently, ETSI published version 1.2.1 of the HbbTV specification in November 2012 [6].

## 5 Version 2.0 of the HbbTV specification (HbbTV 2.0)

Version 2.0 of the HbbTV specification was outlined and written by the HbbTV Association during the years 2013/14. It makes HbbTV up-to-date concerning the used web technology and adds a number of new features. The browser is based on HTML5 and supports various features defined around HTML5. The HTML5 video element can be used in HbbTV 2.0 applications and for backwards compatibility also the "old" video object is available. In addition to the browser update, HbbTV 2.0 includes features like synchronization of broadcast and broadband content, second screen connectivity, integration of CI+ v1.4, TTML [6] subtitling and Push VoD (video on demand). HbbTV 2.0 devices will be backwards compatible in the sense that all HbbTV services deployed in the market on the basis of HbbTV 1.x will be running on the new devices.

HbbTV 2.0 was published by ETSI in 2015. In August 2016, the specification was updated to version 2.0.1 and published by ETSI as ETSI TS 102 796 V1.4.1 (2016-08). HbbTV 2.0.1 adds a number of technical improvements and features to HbbTV 2.0. In September 2018, the specification was updated to version 2.0.2 and published by ETSI as ETSI TS 102 796 V1.5.1 (2018-09) [8]. HbbTV 2.0.2 adds support for HDR (High Dynamic Range), HFR (Higher Framerates) and NGA (Next Generation Audio). The HbbTV Consortium is also working on a Test Suite for HbbTV devices and continuously updates it to the latest specification versions.

---

[6] Timed Text Markup Language, a W3C Recommendation, ref. http://www.w3.org/TR/ttaf1-dfxp/ used in the version as defined by EBU-TT, ref. https://tech.ebu.ch/ebu-tt.

## 6 Applying HbbTV to non-DVB broadcasting systems

HbbTV, as it is today, explicitly specifies integration with DVB broadcasting systems. But the whole concept of HbbTV including the browser specification and the application handling can be applied also to other broadcasting systems. This allows for adopting HbbTV on a global basis and using it as technical basis for huge markets for applications and devices. Currently the integration of HbbTV with the ISDB-T broadcasting system is being tested [9].

# References


[1] ETSI TS 102 796 V1.1.1 (2010-06), Hybrid Broadcast Broadband TV.

[2] CEA-2014 revision A, "Web-based Protocol and Framework for Remote User Interface on UPnP™ Networks and the Internet (Web4CE)".

[3] Open IPTV Forum Release 1 specification, volume 5 – "Declarative Application Environment" v1.1 Available from http://www.oipf.tv/downloads.html.

[4] ETSI TS 102 809, "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in hybrid broadcast / broadband environments"; V 1.1.1.

[5] Description of HbbTV 1.5 from www.hbbtv.org. The HbbTV Specification V1.5 (updated version 1st August 2012) can be downloaded as http://www.hbbtv.org/pages/about_hbbtv/HbbTV-specification-1-5_Aug2012.pdf.

[6] ETSI TS 102 796 V1.2.1 (2012-11), Hybrid Broadcast Broadband TV. http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.02.01_60/ts_102796v010201p.pdf.

[7] Ralf Pfeffer, Sebastian Siepe, Benedikt Vogel, Roberta Campo, Cristina Párraga Niebla, "HbbTV: a powerful asset for alerting the population during a crisis", in Proceedings of the 2013 NEM Summit, October 28-30, 2013 – Nantes, France.

[8] ETSI TS 102 796 V1.5.1 (2018-09). https://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.05.01_60/ts_102796v010501p.pdf.

[9] GLOBAL ITV Project / www.globalitv.org / EU–Brazilian strategic partnership project to develop a globally interoperable iTV platform, enabling the coexistence, interoperability and convergence of existing iTV standards. Funding for this project is granted by the European Commission within the European Union's Seventh Framework Programme (FP7/2007-2013, call FP7-ICT-2013-10.2) under grant agreement n° 614087 and by the Brazilian Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under grant CHAMADA MCTI/CNPq Nº 13/2012 and project number 490088/2013-9.

# Annex 2

# The Hybridcast system

## 1 Introduction

Hybridcast is an integrated broadcast-broadband (IBB) system that uses HTML5 standardized in Japan. The system offers a variety of services through a combination of broadcast and broadband telecommunication resources and features. The latest version considers most of the requirements defined in Recommendations ITU-T J.205 and ITU-R BT.2053. To address these requirements, the specifications define system model, application model, application control signals, receiver behaviours, additional application programming interfaces (APIs), etc. The specifications also define a mechanism for collaborating with companion devices such as tablet computers. In addition, ARIB STD-B62 defines extensions to support interactive Ultra-High-Definition Television (UHDTV) using the MPEG Media Transport (MMT).

As of April 2022, more than 18 million Hybridcast capable receivers are in the market in Japan.

## 2 Specification structure

Hybridcast system is defined by three specifications:

– IPTVFJ STD-0010 “Integrated broadcast-broadband system specification”;

– IPTVFJ STD-0011 “HTML5 Browser specification”;

– ARIB STD-B62 “The second generation multimedia coding scheme for digital broadcasting”.

IPTVFJ STD-0010 defines system model, application model, application control signals, transport protocols, VOD behaviour, monomedia coding, and receiver functions. IPTVFJ STD-0011 defines HTML application structure, behaviour and syntax of elements, and additional objects and APIs.

Specifications for IPTVFJ are publicly available on IPTV Forum Japan website, at:

http://www.iptvforum.jp/en/download

ARIB STD-B62 defines Hybridcast application environment for UHDTV and describes how the Hybridcast application environment works with MMT. ARIB STD-B62 specification is available at:

http://www.arib.or.jp/tyosakenkyu/kikaku_hoso/hoso_kikaku_number.html

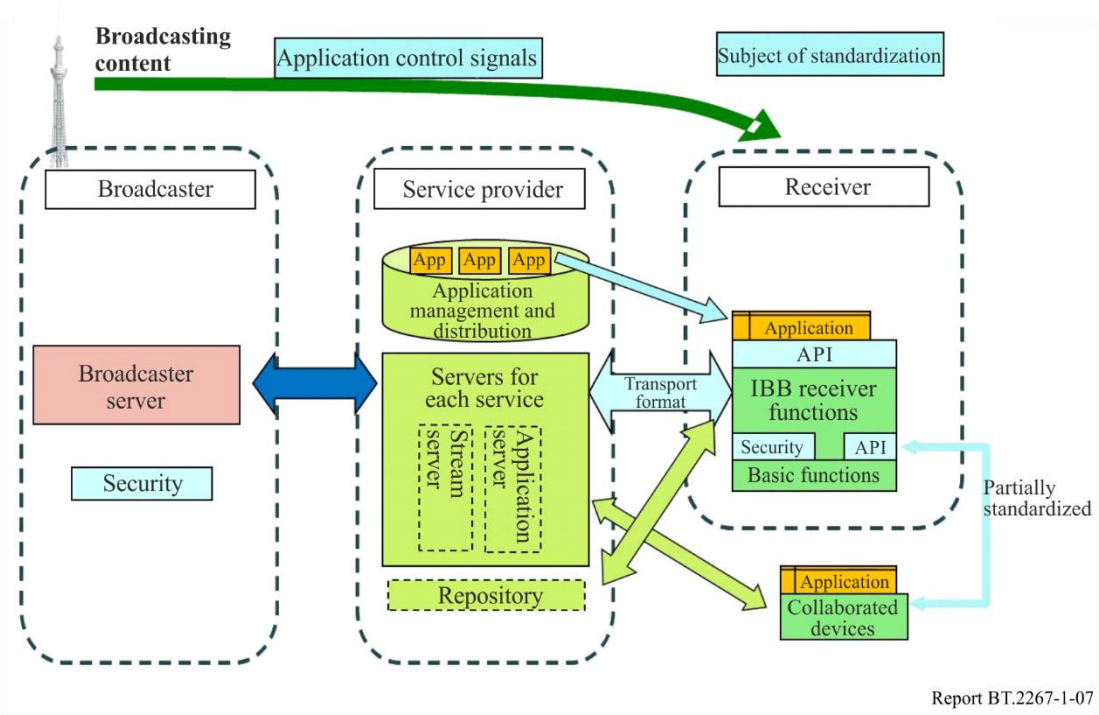
IBB system may be required to transmit application control signals over a broadcast channel in order to control application’s lifecycle, execution conditions, and permissions for access to resources. Vol. 4 of ARIB STD-B24 “Data coding and transmission specification for digital broadcasting” and ARIB STD-B62 define the formats and transmission scheme for application control signals of Hybridcast applications.

## 3 System model

The conceptual overall system model of Hybridcast is shown in Fig. 1-7. This model is intended to expand the services easily by allowing third parties other than broadcasters to participate in the service chain as service providers who develop and distribute applications. When the broadcaster works also as the service provider, or when the broadcaster has some control to the service providers for the service, the form of offering services is on the broadcaster centric scenario.

FIGURE 1-7

**Overall system model**




Roles of Broadcaster, Service provider and Receiver are defined as follows:

1) **Broadcaster**

The broadcaster provides digital broadcasting signals, and metadata and video content to service providers. In the broadcasting signal, application control and permission information can be transmitted, which notifies available applications, information to launch them, and permission to access to broadcast resources. Upon contract, the service providers can obtain metadata and video content from the broadcaster to use or offer it to the end users.

2) **Service provider**

The service provider is the key figure who provides services with this system. The service provider produces/distributes content and application to provide services, and operates servers to enable each service. It is possible to provide link information to other service providers. In addition, APIs that are implemented on servers for access from receivers shall not be specified in the specifications because such APIs are service specific. The service provider can freely set and utilize them.

Some service providers may provide repository function. The repository registers applications to be distributed and provides lists of applications according to inquiries from receivers. Use or access to Repository is considered mainly for non-broadcast-oriented managed applications.

3) **Receiver**

The receiver provides functions to execute Hybridcast applications, make presentation controlled by the application, and interact with users and/or other devices as well as to receive and present broadcasting content. The receiver provides standardized functions and APIs for execution of the applications.

The companion device collaboration function of receivers allows Hybridcast application to interact with companion devices such as tablet computers. Such devices are assumed to be connected to the receiver within the home network.

# 4 Application model

## 4.1 Application type

In the Hybridcast system, there are four types of applications according to the relationship with broadcasting services.

– **Broadcast-oriented managed applications**

Applications which are controlled by the application control signal in the broadcast signal. While the applications are launched or terminated by the control of broadcaster, the applications are allowed to access broadcast resources based on permission information in the control signals.

– **Non-broadcast-oriented managed applications**

Applications that are not controlled for launch, sleep and termination by the application control signal in the broadcast signal but still authenticated by broadcasters by means other than broadcast signal. Permission to access to broadcast resources is given by application authentication, etc.

– **Broadcast-independent managed applications**

Applications that are launched when broadcast services are not selected but are still authenticated by broadcasters. The applications are not allowed to access all the broadcast resources. The applications are launched or terminated by means other than broadcaster signals. If the broadcast signal is being received before start-up, the applications can be launched based on the instruction of the broadcast signals.

– **Unmanaged applications**

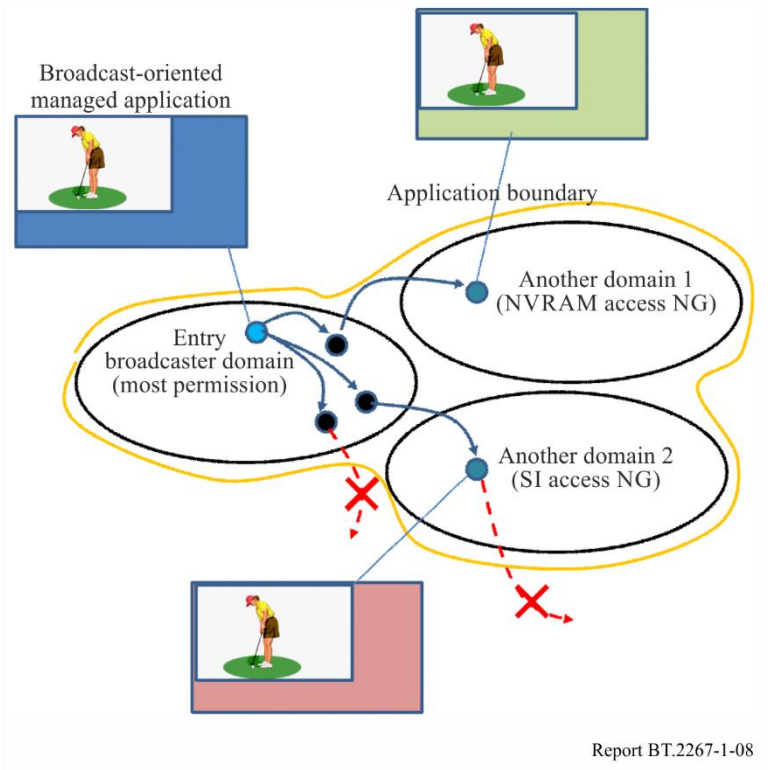
General name for applications that do not apply to either of the above two types of applications. Access to broadcast resources is strictly prohibited.

## 4.2 Application boundary and permission

As described in § 2, third-party made applications, or a part of applications, can be used in the Hybridcast system. To maintain interest of stakeholders or consistency of the services using broadcast-oriented managed, non-broadcast-oriented managed applications, and broadcast-independent managed applications information of accessible network domains with permission information of each domain, called application boundary, is given to each application. The application can be downloaded from or jump to the given domains only. Permission information represents allowable access to broadcast resources for each domain where an application document is downloaded from. Figure 1-8 shows this concept.

FIGURE 1-8

**Concept of application boundary**




## 5 Application control signal

### 5.1 Application Information Table (AIT)

In Vol. 4 of ARIB STD-B24 and ARIB STD-B62, the Application Information Table (AIT) is defined to represent application control information. AIT carries information of execution state of the application such as launch, sleep or termination, application boundary, URL and transport protocols to download the application, etc.

There are two notations for AIT. MPEG Section form of AIT is mainly intended for delivery of AIT over broadcast channel. XML notation of AIT is intended for delivery of AIT over broadband telecommunication network or DSM-CC carousel in broadcast channel.

### 5.2 Coexistence with data broadcasting

In Japan, data broadcasting services are common and widely used. And it is supposed that the Hybridcast receiver provides data broadcasting environment (BML browser) as well to support existing digital broadcasting services. When the broadcaster provides both data broadcasting content and Hybridcast services at the same time in the same channel, it is important for the Hybridcast receiver to determine which of the two should be launched.

ARIB STD-B24 defines a method to control invocation priority for data broadcasting content and Hybridcast application. When data broadcasting is prioritized, BML browser should start regardless of application execution information in AIT. BML browser in Hybridcast receiver provides a new function to switch to Hybridcast applications, which allows data broadcasting content to switch to Hybridcast applications. In this case, an AIT file in XML notation will be retrieved over broadband telecommunication network to control Hybridcast applications. URL of the AIT file is given as

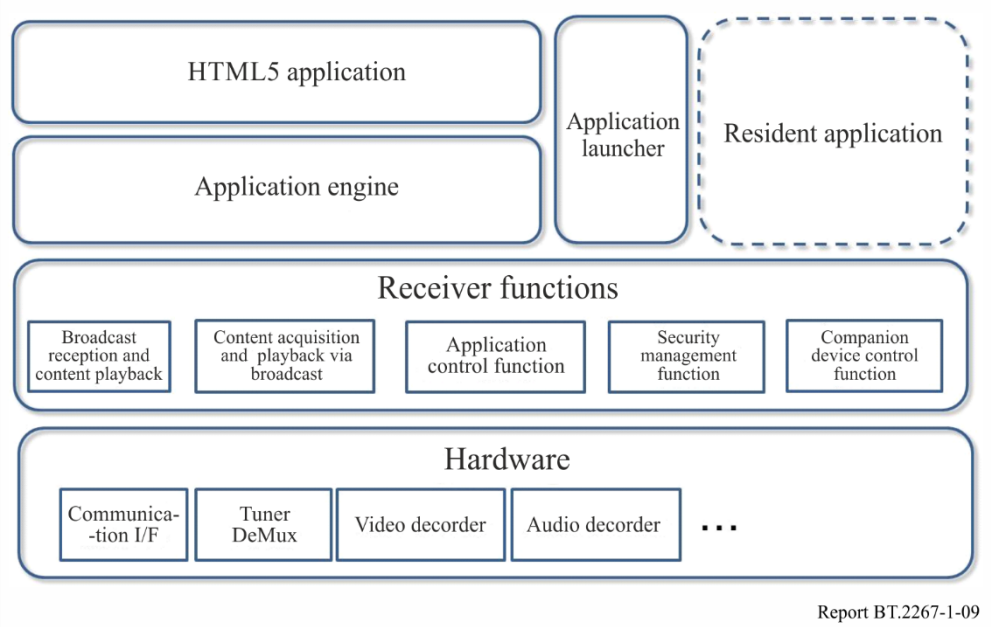
an argument to the function in BML. When data broadcasting is not prioritized, a Hybridcast application can be launched directly in accordance with the broadcast delivered AIT.

## 6 Hybridcast receiver

Figure 1-9 shows a structure of Hybridcast receiver. The functions to receive and present broadcasting programme and/or content are the same as those for existing digital broadcasting services. The processing of Hybridcast applications is also similar to data broadcasting content, and then the structure is similar to that defined in Recommendation ITU-R BT.1889. Some function blocks are specific to Hybridcast for security, application management, and handling of content element and functional binding between a receiver and servers.

FIGURE 1-9

**Structure of Hybridcast receiver**




Following are Hybridcast system specific functional blocks:

– **Content acquisition and playback via broadband**

  Functions that access the video-on-demand content servers and play it.

– **Application control**

  Functions that work with application engines mainly regarding broadcast-oriented managed, non-broadcast-oriented managed and broadcast-independent managed applications in accordance with the application control information. These functions also control/manage lifecycles of applications.

– **Application engine**

  Functions that execute Hybridcast applications. HTML5 browser is used in Hybridcast system.

– **Companion device collaboration control**

  Functions that discover, connect to, and manage collaborative activities with external collaborated devices.

– **Security management**

Functions that apply restrictions to application behaviour as necessary according to instructions specified by broadcast signals, etc.

– **Application launcher**

Navigation functions that are mainly used by end users to select and initiate non-broadcast-oriented managed applications.

## 7 HTML5 environment and additional APIs

IPTVFJ STD-0011 defines HTML5 environment with additional APIs for Hybridcast applications. As a base environment, IPTVFJ STD-0011 employs several W3C Recommendations and Working Drafts such as HTML5, Media Queries, Device Adaptation, CSS Font Module Level 3 and Document Object Model Level 3 Events. It also employs ECMAScript 5th Edition. By employing these W3C materials, the latest ECMAScript specification, and other commonly used W3C specifications such as CSS Animations, an HTML5 browser will work in a modern way, i.e. flexible behaviour and presentation can be controlled by scripting programmes as parts of HTML applications. However, to make an HTML application be a Hybridcast application, it is required to add APIs to control the behaviour of an HTML application itself, to access to broadcast resources, and to call receiver functions. Table 1-1 shows major additional objects and elements.

TABLE 1-1

**Major additional objects**

| | |
|---|---|
| Application object | An object which represents an application itself. This object is used to manage execution state of an application and to invoke another application. |
| ApplicationInformationTable object | An object which represents application control signals. An application can obtain boundary and permission information delivered by application control signals by calling this object. |
| ReceiverDevice object | An object which represents functionalities of a Hybridcast receiver and information managed by a Hybridcast receiver. This object provides functionalities for tuning, access to receiver specific information such as receiver identification and system information, detection of current broadcasting programme, synchronized presentation control, and presentation and data acquisition control of closed caption, etc. |
| EIT related objects | Objects which handle broadcasting programme schedule information. |
| StreamEventTarget object | An object to handle stream events delivered over broadcast channel. This object provides functions to add and remove event listeners, to access to timer, and to handle stream event update. |
| BMLCompatObject object | An object to access to a part of functionalities of BML browser such as non-volatile memory. |
| Companion device collaboration related functions | As a part of ReceiverDevice object, functions to instruct initial URL of an application to run on companion devices and to send/receive messages to/from the devices are defined. |
| BroadcastVideoObject element | An object element with additional type to represent broadcast video instead of video element in HTML5. This object is defined to satisfy broadcast related requirements such as continuous presentation of broadcast video even during loading HTML document, control of z-order alignment of broadcast video and graphics with alpha blending, and mitigation of the risk of canvas element in HTML5 for content security. |

## 8 Companion device collaboration

Manipulating a Hybridcast application with a remote controller is a common way for end users to control the application. External devices such as tablet computers or smartphones – rather than a dedicated remote controller – will dramatically improve the ease of manipulation and the effectiveness of the presentation of the applications. For this purpose, Hybridcast specifications define a mechanism and the APIs needed to adopt such devices, called companion devices, within a home network.

To collaborate with companion devices, a Hybridcast receiver or a Hybridcast application needs to discover the devices on a home network. Moreover, connecting and communicating with the devices require management. Mechanisms for the discovery and the communication vary depending on the circumstances. Discovery and communication with low level entities, for instance, typically require careful consideration and detailed application control, whereas a relatively simple definition is needed with high level entities or functions.

FIGURE 1-10

**A mechanism for companion device collaboration**

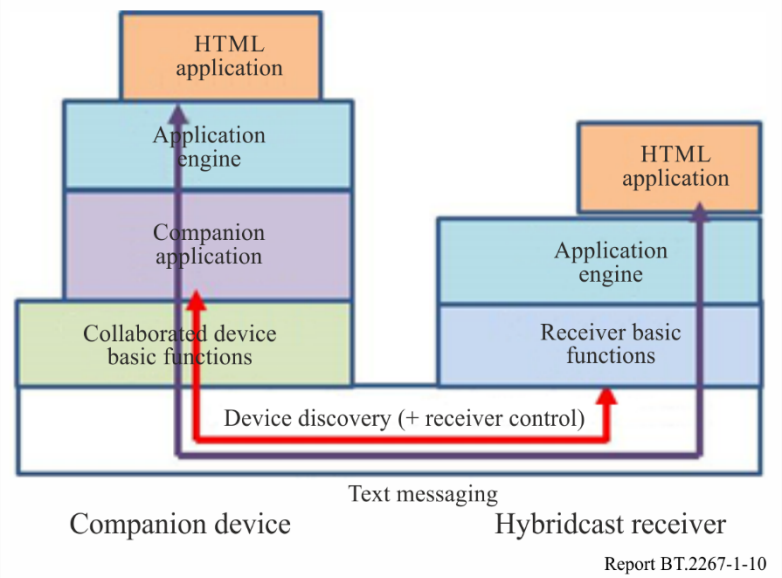



Figure 1-10 illustrates the mechanism for companion device collaboration by the communication between HTML applications on a receiver and a companion device. Because this is a case of discovery and communication between high level entities, relatively simple APIs in the Hybridcast specifications will suffice to enable the communication. The companion application on a companion device is dedicated software for companion device collaboration, and it is developed and delivered by the receiver's manufacturer. The companion application works for discovering the device, managing the connection and communication, and launching the application on the companion device. In a typical implementation, it will also include a user interface as a remote controller. A Hybridcast receiver seeks companion devices within the connected home network. A companion application responds to a device discovery action for the Hybridcast receiver. When they recognize each other, a Hybridcast receiver and a companion application can successfully communicate. The Hybridcast application on the Hybridcast receiver provides the initial URL for the HTML application on a companion device through the API in the 'ReceiverDevice' object. A companion application loads the HTML application in accordance with the given URL. The loaded HTML application for a companion device runs on an application engine that is developed by the receiver's manufacturer. The application engine is normally delivered with a companion application. Once an

HTML application on a companion device has been launched, communication by text-based message is possible between the Hybridcast application on a Hybridcast receiver and an HTML application on the companion device.

FIGURE 1-11

**Entities and their relationship for companion device collaboration**

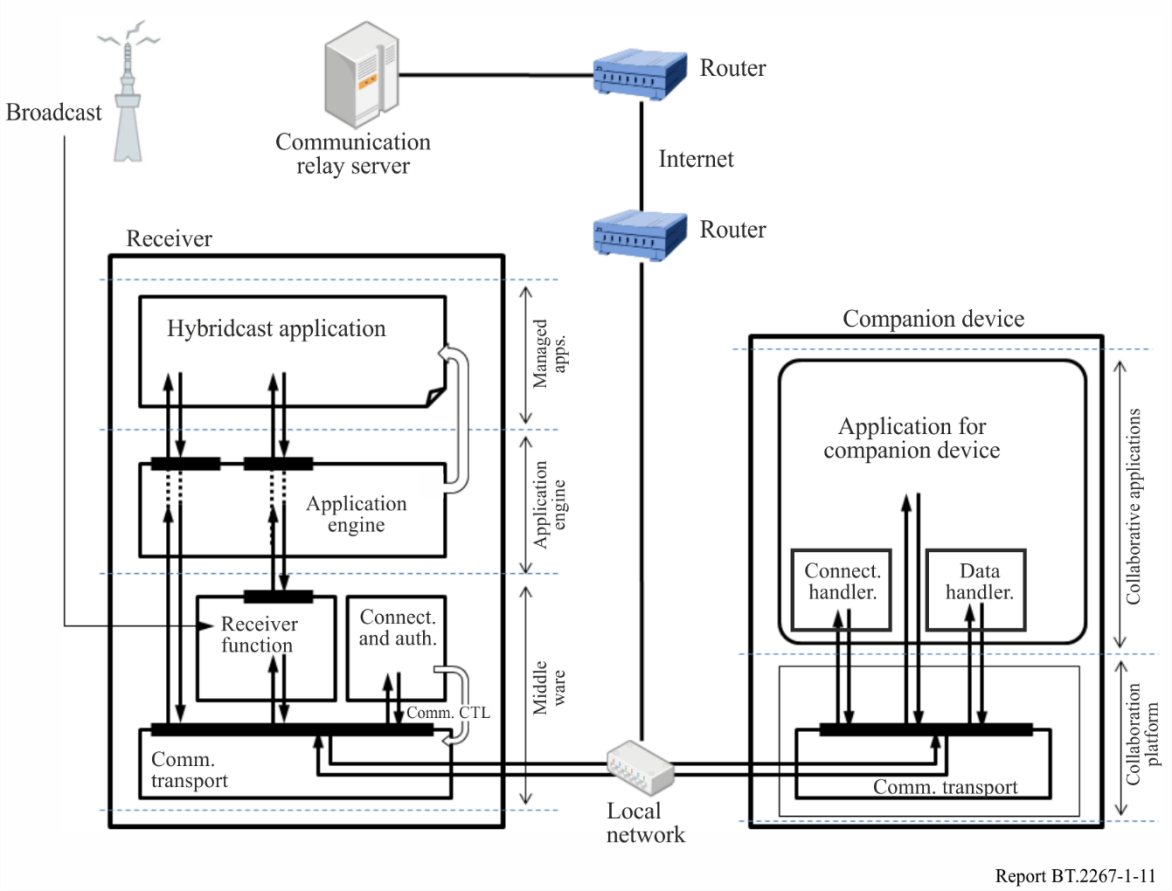



Figure 1-11 illustrates a detailed companion-device collaboration model, as defined in the Hybridcast specifications. This model is defined for various interaction scenarios, including the interaction between an application on a companion device and a function on the receiver, the interaction between a companion device and a receiver via a communication-relay server that is located on the Internet, etc. However, the protocols for communication and device discovery may be different in each scenario. Furthermore, as mentioned, access to low-level functions requires careful consideration in particular for rights management and viewer safety. In addition, access to low-level functions should be harmonized with other services such as those offered by manufacturers. Considering this situation, the actual protocols used and the permission control will be defined in future versions.

## 9 Advanced VOD Playback

It is important for viewers to have the same user experience when they watch a programme with a Hybridcast application in a time-shifted manner, that is, with VOD or recordings. For this reason, the Hybridcast specifications define a mechanism to initiate an application when playing back VOD content or recordings. When a receiver ‘tunes to’ VOD content, and if a VOD server sends an application control signal, the application will be initiated as if the programme is being presented live. The same approach is adopted for recordings.

A specific problem in playing back VOD content or recordings is the manner by which it handles trick-play (i.e. 'fast forward,' 'fast backward,' and 'slow motion'). With trick-play, the timing of the content does not progress linearly. Trick-play will lead to the unsynchronized presentation of a Hybridcast application that is running. To permit an application to recognize the playback position in time, events for entering and deviating from the normal playback speed are defined, along with an API to detect the time position.

Further, considering that companion device collaboration enables the playback of VOD content on various devices, support for MPEG-DASH maximizes the compatibility of audio and video content.

## 10 Non broadcast-oriented managed application

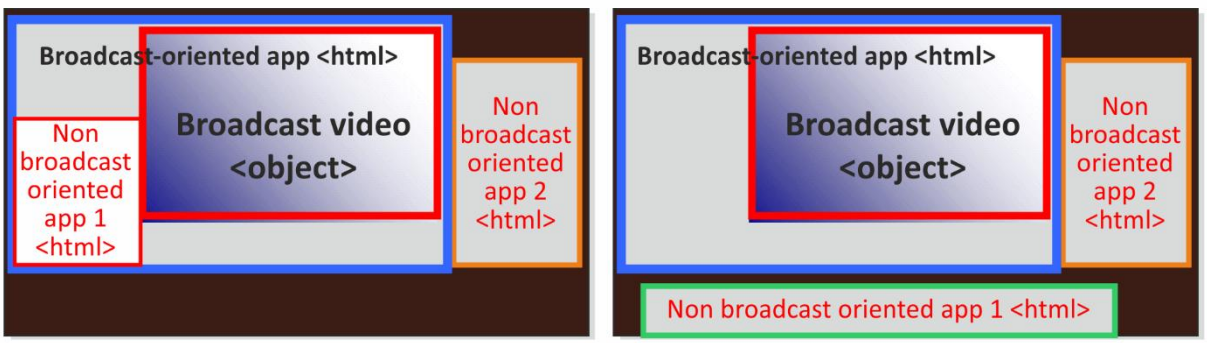
Non broadcast-oriented managed applications are independent from broadcast channels and can be initiated and stopped at any time by viewers. Applications of this type are similar to applications on smartphones or tablets. However, non-broadcast-oriented managed applications support stand-alone applications. The difference between smartphone/tablet applications and non-broadcast-oriented managed applications is that the latter are permitted access to the broadcast resources. That is, non-broadcast-oriented managed applications enable independent services that closely interact with broadcast programmes or channels. It is expected that the introduction of this type of application will lead or motivate web users to watch broadcast programmes.

Applications of this type are required to maintain the integrity of broadcast content. Non broadcast-oriented managed applications are a type of managed application. It is supposed that the broadcasters' consent to the application is required to register it in the repository, an application delivery centre, as shown in Fig. 1-7.

In addition, applications of this type can be managed in terms of their presentation on a display, and with regard to controlling the execution, including access to broadcast resources. For example, unauthorized overlaying of broadcast programmes or disturbances to them are not permitted. On the other hand, the execution and proper presentation of such applications is achievable. In order to ensure this, a management mechanism is defined in the Hybridcast specifications.

FIGURE 1-12

**Example of presentation management for non-broadcast-oriented managed applications**




Case 1

app 1: Overlaying onto broadcast video and broadcast oriented app allowed
app 2: Overlaying not allowed

Case 2

Both app 1 and app 2 are not allowed to overlay

Report BT.2267-1-12

Figure 1-12 provides an example of presentation management on a display for multiple types of applications. In Case 1, a broadcast-oriented managed application is running. This application controls the presentation of the broadcast video. Moreover, app 1 and app 2, both non

broadcast-oriented managed applications, are running. A broadcaster sends the permission signal over the broadcast channel to permit app 1 to overlay onto the broadcast-oriented managed application. However, it does not permit an overlay for app 2. A receiver determines the allocation of applications on a display in accordance with the permission information. Likewise, in Case 2, the allocation of applications on the display is determined by a receiver in accordance with the permission given by a broadcaster, and it prevents overlaying from both app 1 and app 2.

## 11 Broadcast-independent managed application

Broadcast-independent managed applications are launched when broadcast services are not selected but are still authenticated by broadcasters. Applications of this type are launched in roughly two patterns: launched by data broadcasting or other application in the receiver, or launched by external devices.

When attempting to launch a broadcast-independent managed application by data broadcasting or other applications, if the application meets the specifications and the receiver meets the operating conditions, the receiver acts on the application engine to launch the application.

For an application launched by the external launcher of the external device, if the application is approved by the broadcaster or the business entity entrusted by the broadcaster, the application is allowed to be launched.

When application termination or application transition occurs owing to broadcasting service selection, user operation, etc., control and management of the application in operation are terminated.

## 12 Accurate synchronized presentation

In addition to the synchronization control based on a trigger signal, such as a DSM-CC stream event, APIs for accurately synchronizing the presentation are defined. The following two scenarios are considered for accurately synchronized presentations:

- Stream-stream synchronization

  Multiple streams are presented synchronously. This function allows for multi-angled viewing by the application, where video streaming over the internet delivering video images from different angle is accurately presented with a broadcast video image, for example.

- Stream-graphics synchronization

  Graphics are rendered synchronously with the video image. Information related to the objects to be rendered is delivered over the internet. The information contains the location, the time, and the object to be rendered.

To ensure stream-stream synchronization, additional APIs are defined in the Hybridcast specifications to establish a master-slave relationship for media-player objects. For stream-graphics synchronization, APIs are defined to verify the accurate presentation time and to delay the stream as needed.

## 13 UHDTV support

Volume 2 of ARIB STD-B62 defines the Hybridcast application environment for UHDTV. The environment is designed to work with MPEG2-TS and MMT. ARIB STD-B62 defines the presentation model, receiver architecture, protocol stacks, signalling and messaging format, and the modifications to Hybridcast standards owing to differences in the parameters of MPEG2-TS and MMT.

FIGURE 1-13

**Protocol stack defined in ARIB STD-B62 for MMT**

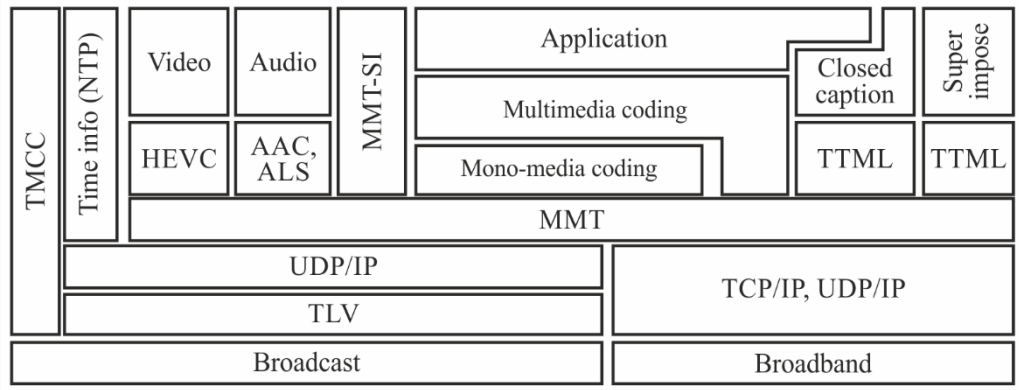



FIGURE 1-14

**Protocol stack defined in ARIB STD-B62 for MPEG2-TS**

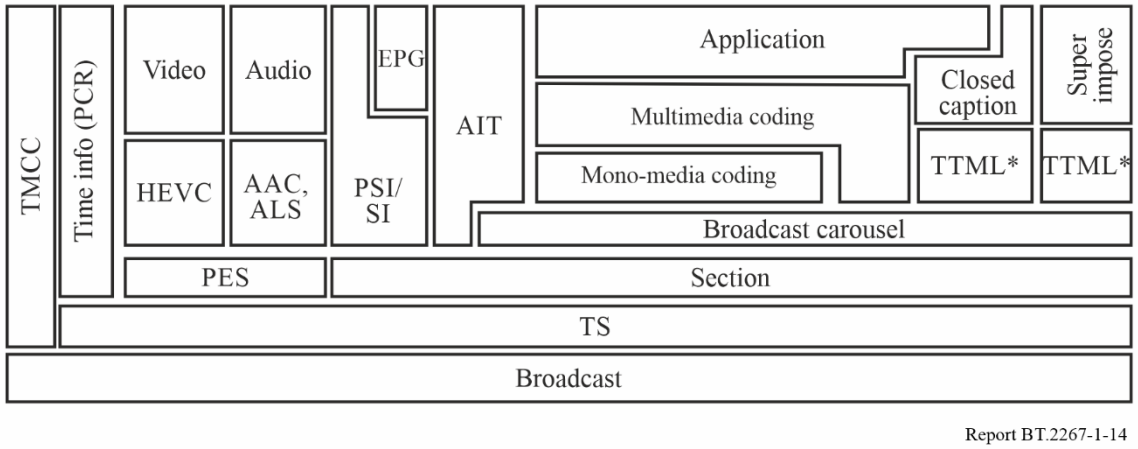



FIGURE 1-15

**Protocol stack defined in ARIB STD-B62 for the broadband**

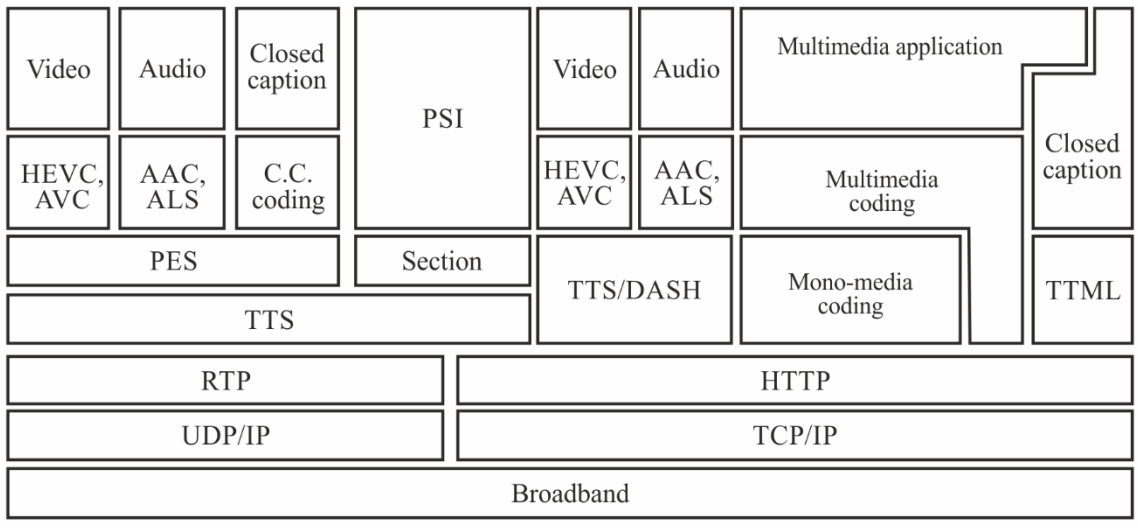


Figure 1-13 shows the protocol stack for MMT. MMT and other related protocol suites in this Figure are defined in ARIB STD-B60, and this enhances ISO/IEC 23008-1 for actual use. ARIB STD-B62 describes how the environment works with MMT. Applications, required resources, and metadata are delivered by file-transfer protocol, and application signalling is delivered by MMT-SI. Closed captioning information is coded in a newly developed format, called ARIB-TTML, as defined in Vol. 1, Part 3 of ARIB STD-B62, and it is delivered by file-transfer protocol as well.

Figure 1-14 shows the protocol stack for MPEG2-TS. This is almost the same as that for existing digital broadcasting system in Japan. There are small differences around closed captions and super impose, and codecs.

Figure 1-15 shows the protocol stack for the broadband which is commonly used with MMT and MPEG2-TS. The protocol stack shown in Figs 1-13 and 1-14 focuses on 'push' delivery, and that in Fig. 1-15 focuses on 'pull' delivery.

FIGURE 1-16

**Reference receiver architecture defined in ARIB STD-B62**

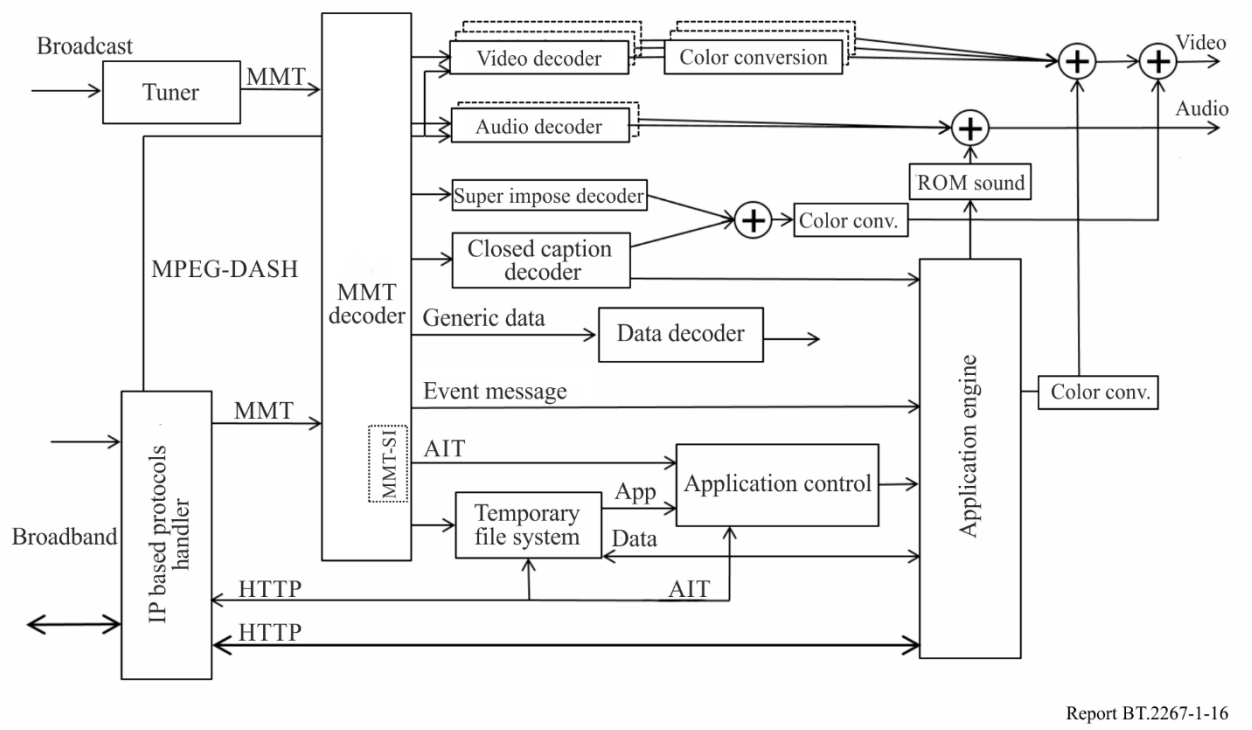



Figure 1-16 charts the reference receiver architecture. Closed captioning data can be retrieved by the multimedia application. The application control signal (AIT) comes from various paths to a receiver in accordance with the application type. The ROM sound is a built-in sound, such as a chime, in a receiver. A colour-space conversion capability is required in a receiver because broadcast video from UHDTV will comply with Recommendation ITU-R BT.2020, whereas video streaming over broadband may comply with Recommendation ITU-R BT.709. Colour space conversion is also needed in order for the graphics to be compatible with the video images.

## 14 Service examples

Some service examples enabled by Hybridcast specifications are described in this clause.

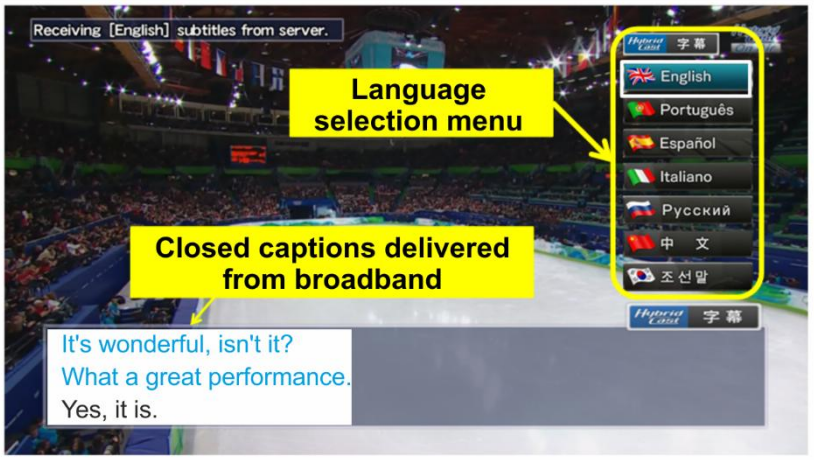
### 14.1 Multilingual closed caption

According to Japanese digital broadcasting standard, closed caption can be provided for up to two languages within a broadcast channel. Two languages may not be enough for some people including minority or foreign travellers. If a viewer would like to watch closed caption in a language not delivered over the broadcast channel, additional closed caption text in a preferred language may be obtained through broadband telecommunication network. The closed caption server feeds caption data in the requested language and corresponding time information. The Hybridcast application renders caption texts in accordance with given time information.

Figure 1-17 shows an example image of this service. In Fig. 1-17, seven languages are available. A viewer selects the preferred language by using a remote controller, and closed caption in selected language is on the screen.

FIGURE 1-17

**Multilingual closed-caption**




Report BT.2267-1-17

### 14.2 Social TV

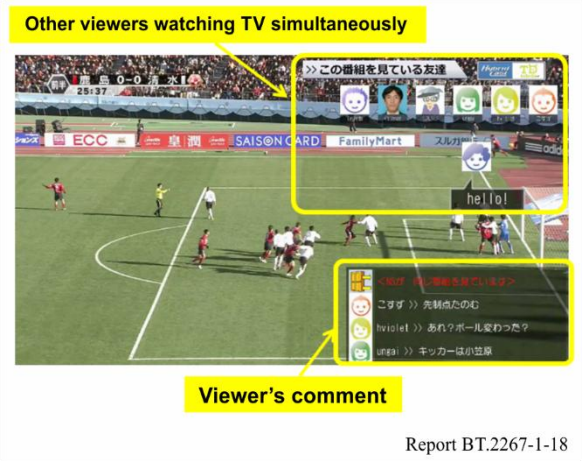
One of the highlights of TV viewing is to spend time and discuss with family or friends while watching TV programmes, especially live broadcasting programme. It is quite natural to do so when a family shares the single TV set and watches the same programme together. By integrating with a Social Networking Service (SNS), Hybridcast enables people who are not in the same location to share the feelings as if they were nearby. It is called Social TV.

Figure 1-18 shows a chat application example. It lists friends who are watching the same programme on top-right of the screen by icons. On the bottom-right, chat messages are displayed.

The Hybridcast application manages the login status of SNS and watching status of the programme. The application obtains the ID of programme from the receiver. The application then sends the ID of programme to the server for the SNS. The friends or family who are logged in and have registered the same ID of programme become chat members. Thus, Social TV service can create a virtual space for TV experience and overcomes physical distance among friends or family.

FIGURE 1-18

**Social TV**




## 14.3 Language study

Providing an educational broadcasting programme is one of the important elements of a public service. A language study programme is a typical example of this. Hybridcast can provide a language study service with a secondary screen device (Fig. 1-19), which contributes to an efficient and effective education. An application on a tablet is used to ask questions about conversation in the programme. Viewers can answer the questions using touch controls.

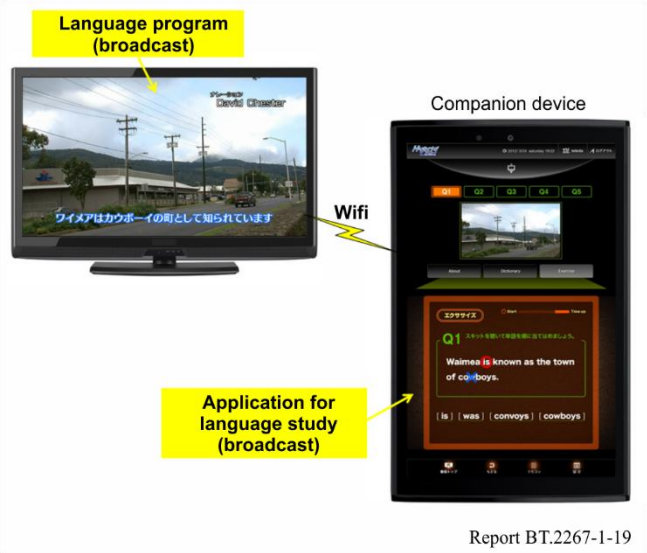
Stream event within a broadcast channel indicates a moment to progress to a next step of lessons. For example, a broadcaster sends the stream event when asking a question. Once the Hybridcast application on a Hybridcast receiver receives the stream event, it tells the application on the tablet to acquire data of the question over the telecommunication network and to start asking it.

Such a mechanism and user interface of a tablet may introduce viewers to new experiences of language learning. Especially, the touch interface makes learning fun and viewers may not get tired quickly.

Use of the companion device as an interactive text book can be applied to other subjects of education. It will allow reviewing the matters of wrong answers later very easily, and help effective learning.

FIGURE 1-19

**Language study**




## 14.4 Multilingual caption with companion devices

As described in the previous clause, secondary screen devices, or companion devices, can be integrated into the Hybridcast services. An application that uses a companion device for multilingual closed caption was developed to evaluate the effectiveness of use of the device. To support a variety of languages, closed-caption data are delivered over a broadband channel. The use of the device enables language selection without disturbing the content on the primary screen. For animation programmes, introducing the captions in balloons near the relevant objects on the screen will provide ease to the viewers in understanding the programmes. Figure 1-20 shows example images of a primary screen in various languages with regard to this application.

FIGURE 1-20

**Multilingual closed captions in balloons on a TV screen**




Japanese




English



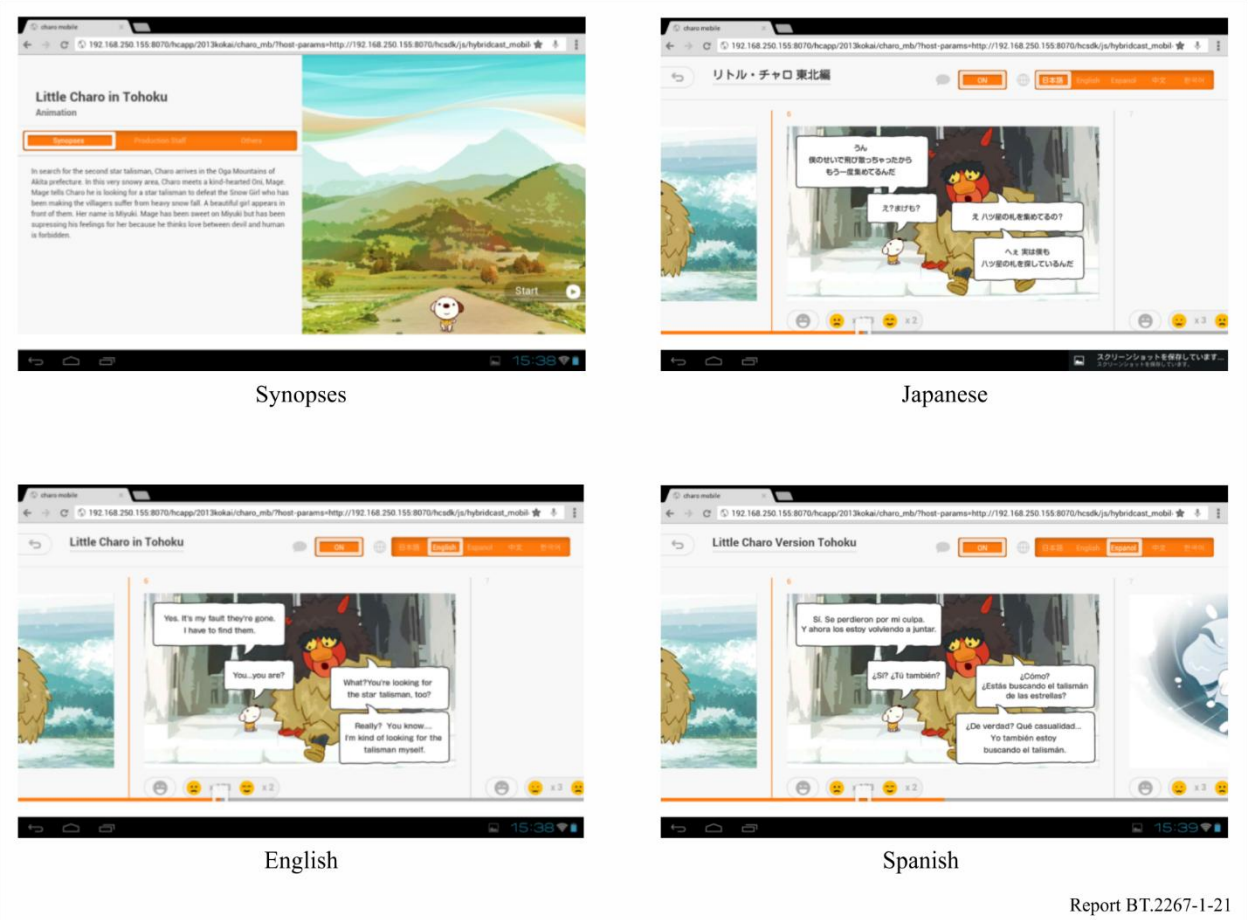

Spanish

Report BT.2267-1-20

In addition, the companion device can be used to improve accessibility to the content. Synopses of the content and presentation of a representative picture of each scene with captions on the device will facilitate understanding of the story by the viewers. The presentation of the picture on the device changes in accordance with the progress of the programme presented on the Hybridcast receiver. Figure 1-21 shows the synopses and a representative scene presentation in multiple languages.

FIGURE 1-21

**Synopses and representative picture of the scene on a companion device**




## 14.5 Non broadcast-oriented managed application

As described in § 10, non-broadcast-oriented managed application can be executed independently from broadcast channel being presented.

FIGURE 1-22

**Presentation management example for non-broadcast-oriented managed application**

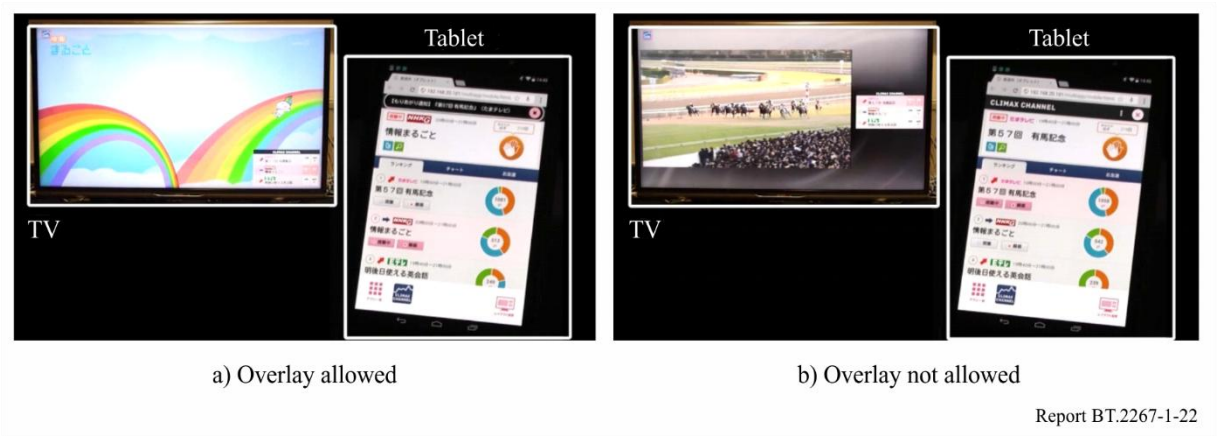



Figure 1-22 provides an example of the behaviour of non-broadcast-oriented managed applications. The non-broadcast-oriented managed application runs on a receiver and communicates with the application on a tablet. The tablet application shows a list of current TV programmes. Similar to the behaviour of smartphone applications, this list periodically updates and refreshes the order of the programmes based on the results of a real-time analysis of comments posted to social-networking services. When a user taps a programme on a tablet, the tablet application sends a message to the application on the receiver to change the channel. That is, the application on the receiver continues to run, regardless of the channel that is broadcast. A broadcaster that allows the presentation of this application on a TV screen in overlay form is shown in Fig. 1-22(a). However, if a user taps a programme broadcast by another broadcaster that does not permit overlaying, the receiver automatically splits the screen to avoid the overlay in accordance with the permission information from the broadcast signal, as shown in Fig. 1-22(b).

## 14.6 Interactive UHDTV

UHDTV, and in particular UHDTV-2 (8K), provides a much higher resolution than HDTV. From a multimedia perspective, the high-resolution display enables variety of information to be seen on the screen at the same time.

FIGURE 1-23

**Example application for UHDTV**

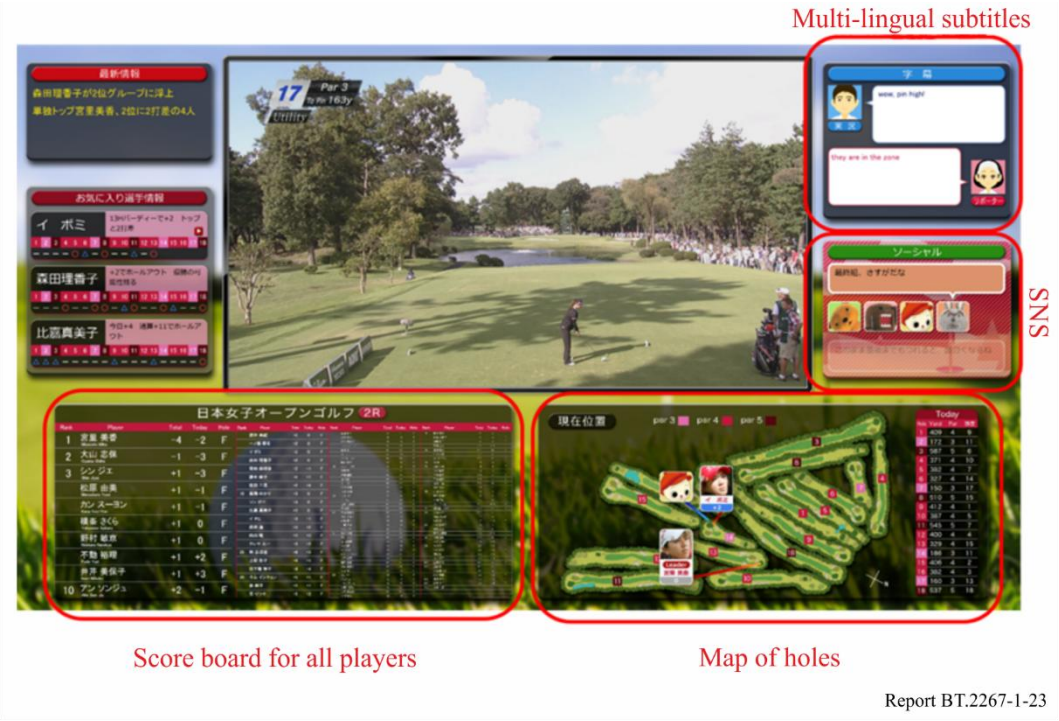



Figure 1-23 shows an example of a golfing application for UHDTV. The high resolution of UHDTV allows for the simultaneous presentation of many kinds of information. In Fig. 1-23, the information that is of interest to most viewers is typically presented. On the bottom left, the scores of all players are listed by changing the font size.

On an 85-inch display, even the smallest characters are still readable. On the bottom right, a map of the greens is displayed, indicating the position of the players in whom a viewer is interested. On the upper right, a multilingual closed caption is available. With this application, a viewer can choose between Japanese, English, and Korean. Beneath the closed caption, comments posted to social networking services by friends who watch this programme are shown. This application can be operated with a companion device, e.g. a tablet, for ease of use, especially when typing a comment to post to a social-networking service.

FIGURE 1-24

**Example application for UHDTV**




Report BT.2267-1-24

The simultaneous presentation of a variety of information will make it easy to share the screen with several people who each want to watch their preferred content. Figure 1-24 illustrates such a case. In the bottom left, VOD allows a viewer to watch a player different from the player in the broadcast programme. In the bottom right, a viewer can enjoy a golf game playing on TV. Of course, a viewer can still enjoy the broadcast programme that appears in the upper centre. With only a slight movement of the eye, viewers who are watching something other than the scene in the broadcast programme can easily keep track of what is going on in the broadcast programme.

As shown in this example application, interactive UHDTV has the potential to satisfy several viewers with a single service, in addition to the high-quality picture offered by UHDTV.

## 14.7 First Hybridcast services

Hybridcast services are planned to be introduced from simple services to sophisticated ones. In September 2013, the first Hybridcast services were launched in Japan. Figure 1-25 shows the appearance of the top-level menu in the Hybridcast services.

FIGURE 1-25

**Top-level menu in the first services**




From the top-level menu, various services, including news, weather, stock market information, and EPG, can be selected. Viewers can select any applications from their own remote controller. Figure 1-26 shows example images when weather forecast is selected from the menu.

FIGURE 1-26

**Screen image of weather forecast**



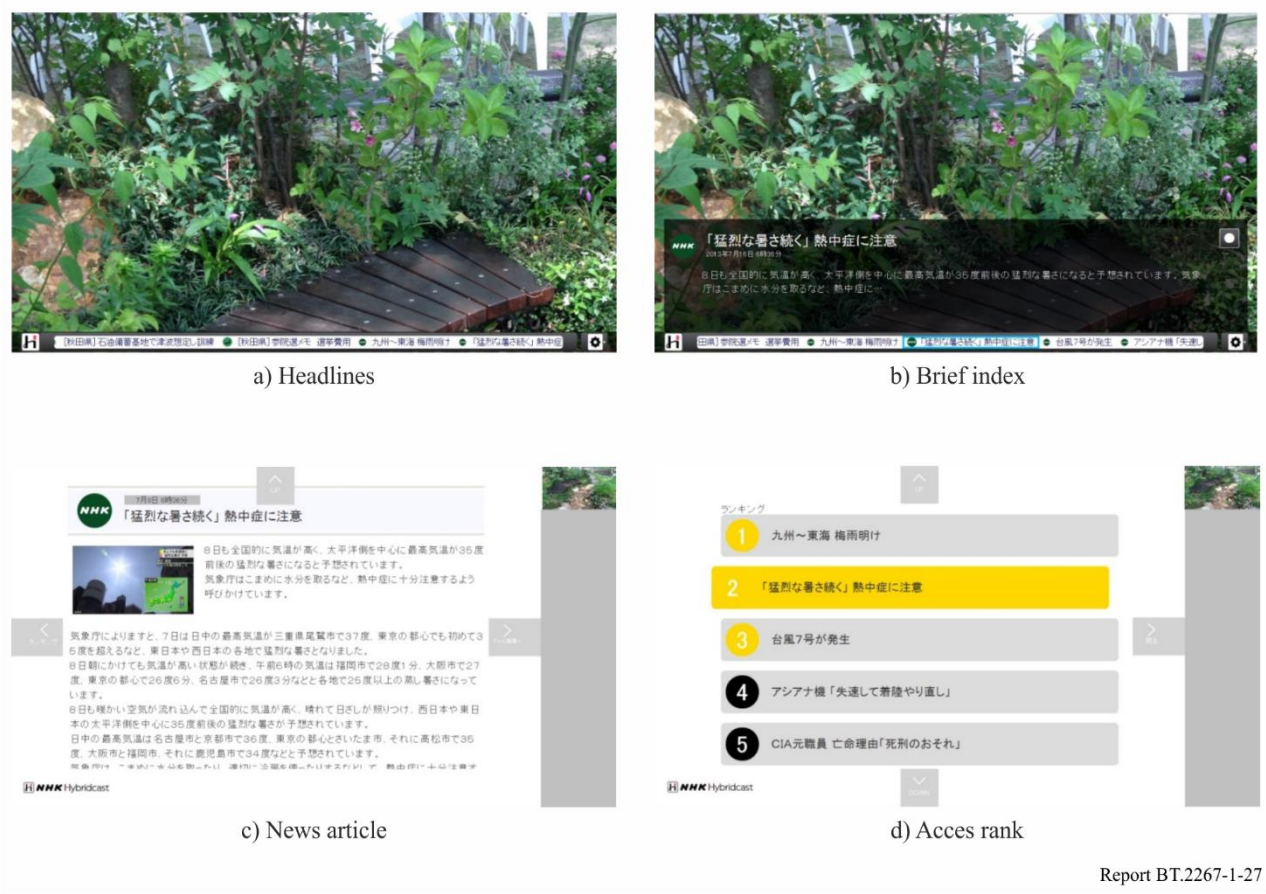

Detailed forecast Current weather

Report BT.2267-1-26

The combination of the broadcast and Web services is a natural feature of the Hybridcast system. For the Hybridcast applications, accessing the Web sites to retrieve information is easy because the functionality of Hybridcast applications is based on HTML5. Figure 1-27 shows example presentations of the Hybridcast news application that retrieves news articles and headlines from Web services.

FIGURE 1-27

**Information retrieval from websites by the Hybridcast application**




a) Headlines b) Brief index

c) News article d) Acces rank

Report BT.2267-1-27

Information on the news headline, brief index, and news article, as shown in cases (a) to (c), is retrieved by the Hybridcast application from existing news website on the Internet. The application displays the retrieved news headlines at the bottom of the screen by scrolling, as shown in (a). The viewers can configure the application to show the headlines on the basis of the preferred genres and regions of the news. Each article headline also works as the button to select the article. When one of the headlines is selected, a brief index is displayed, as shown in (b). When the enter key on the remote controller is hit again, the detailed news article is displayed, as shown in (c). Whilst the detailed news article is displayed, information on the access rank of the news items is retrieved and displayed by hitting the left arrow key, as shown in (d). In any case, the same mechanism for information retrieval used on the Web is used for the Hybridcast applications, which also allows broadcasters quick implementation and easy deployment of their services.

With regard to the application implementation, the broadcaster benefits from the use of the data broadcast content as a launcher of Hybridcast applications. In Japan, almost all digital TV receivers provide data broadcast capability, and all Japanese broadcasters have deployed their own data broadcast services since digital broadcast was launched. Along with the development in the Hybridcast specifications, the data broadcast API is enhanced to provide capability to also launch Hybridcast applications. Because Hybridcast-capable receivers provide both Hybridcast and enhanced data broadcast environment, this enhancement of the data broadcast API allows Hybridcast applications to be started by modifying only the existing data broadcast content. This enhancement reduces the need for the broadcasters to install additional Hybridcast signalling facility for simple Hybridcast services and provides opportunity to broadcasters to easily and quickly deploy Hybridcast services. And the API to start data broadcast content defined in the Hybridcast specification allows broadcasters to enhance their services by combination of data broadcast and Hybridcast.

The applications are designed to provide appropriate actions in case of emergency. Broadcasters should present prioritized information as one of their essential roles. In Hybridcast applications, graphic components can be overlaid with the broadcast video, as shown in Figs 1-25 to 1-27. However, in case of an emergency such as a disaster alert, a signal multiplexed in the broadcast stream causes the applications to hide immediately so that the broadcast video is preferentially displayed and notifies the viewers of what is currently happening.

### 14.8 Other services offered

In addition to the services listed above, following services are also offered to viewers.

#### 1) Start-over service

This service offers the opportunity for viewers who have missed the start of broadcast programme to watch it from the beginning by using VOD. This differs from the catch-up service insofar as the start-over service can begin playing the programme before it ends. This service was offered in programmes showing the Winter Olympics in 2014.

#### 2) Multi-view

This service allows viewers to watch the same scene from a variety of angles. The angle to be presented is chosen by the viewer, and the selected video image is delivered over the broadband channel to the companion devices. This service was offered in programmes showing the Winter Olympics in 2014.

#### 3) Programme-related services

These services offer viewers the opportunity to participate in programmes, such as a quiz show, and to post opinions about the programmes and obtain additional information including keywords from the programme. Additional information may be updated in accordance with the progress of the programme. The number of broadcast programmes providing this service is increasing.

## 15 Use cases of companion devices

Companion devices such as tablet computers and smart phones are used for interactive manipulation of the applications and presentation of extended information from TV screen as described in § 8.

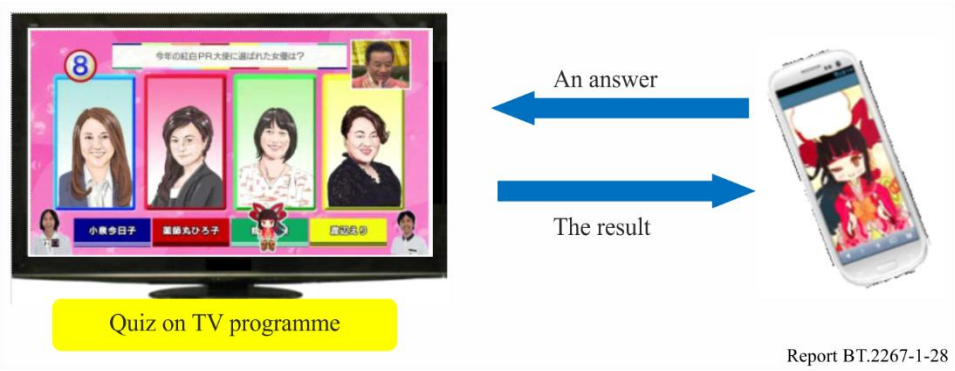
Some use cases of companion devices available on Hybridcast are described in this clause.

### 15.1 Interactive manipulation with broadcast programme

Along with TV programme, a companion device is used for interactive manipulation instead of TV's remote control and for presentation of response as the result of the manipulation. Figure 1-28 shows example presentations of the Hybridcast application for quiz show. User can select the question of the quiz presented on a TV, and the companion device shows the result of the answer to the question which varies by the user's selection.

FIGURE 1-28

**Interactive manipulation with broadcast programme**




## 15.2 Interactive presentation of related information to broadcast programme

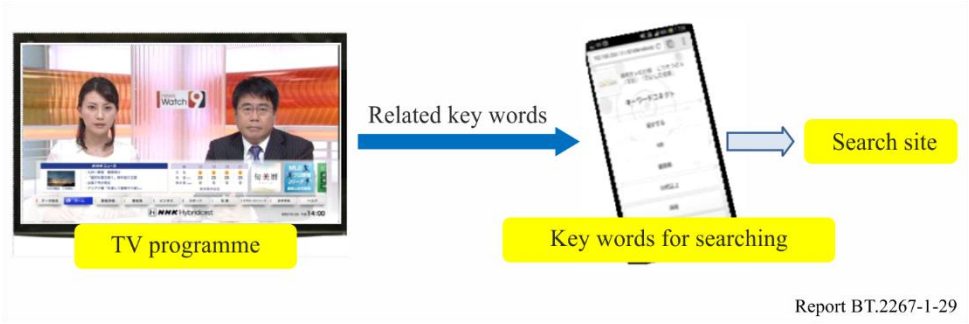
A companion device is used to present detailed information related to a TV programme shown on a TV. Touch-screen interface of a companion device enables comfortable manipulation for a user. An example of this case in Hybridcast is described in § 14.3.

## 15.3 Connection with web sites

Purpose and role of "companion application" is described in § 8. Because the "companion application" is one of the smartphone/tablet applications, it can work with web browser on the device. For example, an HTML application on the "companion application" retrieves a set of the key words specific to the current scene of a TV programme. Once a user selects one of them, meaning of the selected word obtained from the search site is shown on a companion device. Figure 1-29 shows its example appearance. Key words are extracted from closed caption data and EPG data.

FIGURE 1-29

**Example of presentation of words related to on-air programme**




## 15.4 Synchronized multi-screen presentation of video streams

A companion device is capable of presenting streaming video. Selection and playback of streaming video is controlled by Hybridcast application on a TV. For example, streaming video conveys video images of different angle from the images of a TV programme as shown in Fig. 1-30. An application on the TV acquires current system time clock derived from received broadcasting signal by calling an API. The return value of the API represents current playback position in time. The Hybridcast application on the TV notifies such playback position in time to a companion device, and thus the TV and the companion device share the progress of the common time. It enables to synchronize the two videos on the TV and the companion device.

FIGURE 1-30

**Example of synchronized presentation between TV programme and video on companion device**

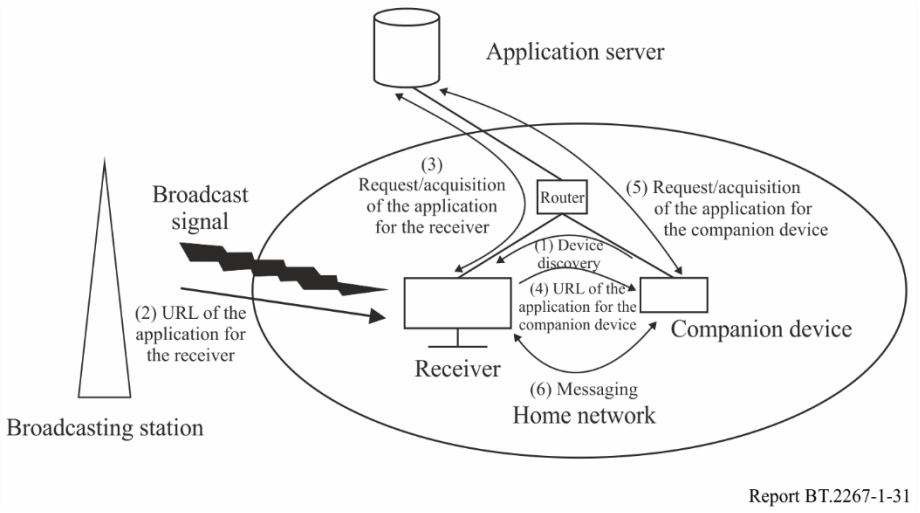



## 15.5 Common companion application

Applications on companion devices described above are implemented on a variety of receivers, which use their own proprietary protocols and user interfaces. Hybridcast defines device discovery protocols and communication protocols between a TV and a companion device. Figure 1-31 shows launching sequence of applications for both a TV and a companion device in a common way. When a companion application on the companion device is started, the companion application searches the receiver over the home network. The application for a receiver instructs the companion application to launch the application for a companion device with URLs of the application and a library of common API. The companion application retrieves and starts the applications from notified URLs. Device discovery protocol is DIAL (DIscovery And Launch protocol) based on UPnP.

FIGURE 1-31

**System configuration example of connecting a receiver and a companion device**




## 15.6 Common companion application to execute an application on the receiver

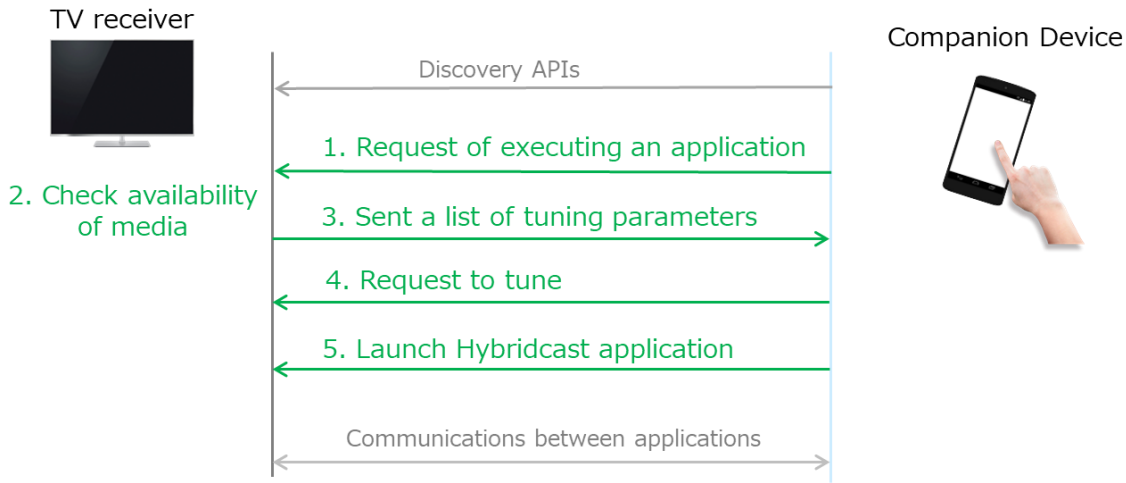
Hybridcast defines protocols to execute an application on the receiver from a companion application on the companion device. The execution sequence and two use cases of such applications are described in this clause.

### 1) Typical sequence for executing an application on the receiver by the companion device

Figure 1-32 shows a typical sequence for executing an application on the receiver by the companion device. When receiving a request of executing an application from the companion device, the receiver checks the availability of a tuner of terrestrial/satellite broadcasting. The companion device acquires a list of available broadcast channels on the receiver and requests the receiver to tune to a selected channel and to launch a Hybridcast application. The receiver then obtains AIT from a server and executes the Hybridcast application according to the AIT.

FIGURE 1-32

**Typical sequence of executing an application on the receiver by the companion device**




### 2) Navigation to a specific TV program by a push notification on the companion device

Figure 1-33 shows an example of navigation to a specific TV programme by a push notification on the companion device. Broadcaster A provides an application concerned with Programme X (Application P) for the companion device. Application P has a function to receive a push notification informing “Programme X is now on air” and a function to execute a broadcast-oriented application on the receiver. The sequence of operation is as follows:

1) Application P, which is connected to the receiver by the device discovery described in § 14.5, is pre-installed on the companion device.

2) When a user is viewing a programme by Broadcaster B, Application P receives a push notification informing “Programme X is now on air”.

3) The user executes Application P on the companion device by tapping the notification and selects the function to view Programme X on the receiver.

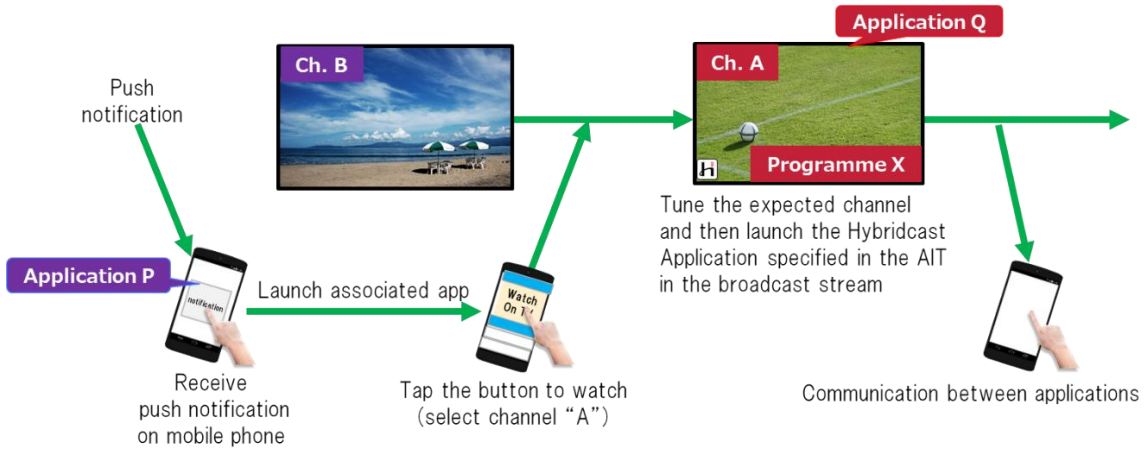
4) Application P requests the receiver to execute a Hybridcast Application Q concerned with Programme X.

5) After receiving the request, the receiver tunes to Broadcaster A and executes Application Q.

6) Application Q sends a URL concerned with Programme X to Application P. Application P on the companion device receives the URL and presents the Web site linked by the URL.

FIGURE 1-33

**An example of navigation to a specific TV programme by a push notification on the companion device**




### 3) On-demand services on the receiver controlled by an application on the companion device

Figure 1-34 shows an example of on-demand services on the receiver controlled by an application on the companion device. Broadcaster C provides Application R for the companion device, which provides OTT services for catch-up programmes. Application R has a function to play an on-demand programme on the companion device and a function to execute a broadcast-oriented application to play the on-demand programme on the receiver. The sequence of operation is as follows:

1) Application R, which is connected to the receiver by the device discovery described in § 15.5, is pre-installed on the companion device.

2) The receiver is switched-off (standby).

3) When a user selects a programme in the list of available on-demand programmes on Application R, Application R requests the receiver to tune to Broadcaster C, to execute a Hybridcast Application S for playing on-demand programmes on the receiver and to play the selected programme on the application.

4) When receiving the request from Application R, the receiver turns on by itself and tunes to Broadcaster C. The receiver then executes Application S on the receiver and plays the selected programme on Application S.

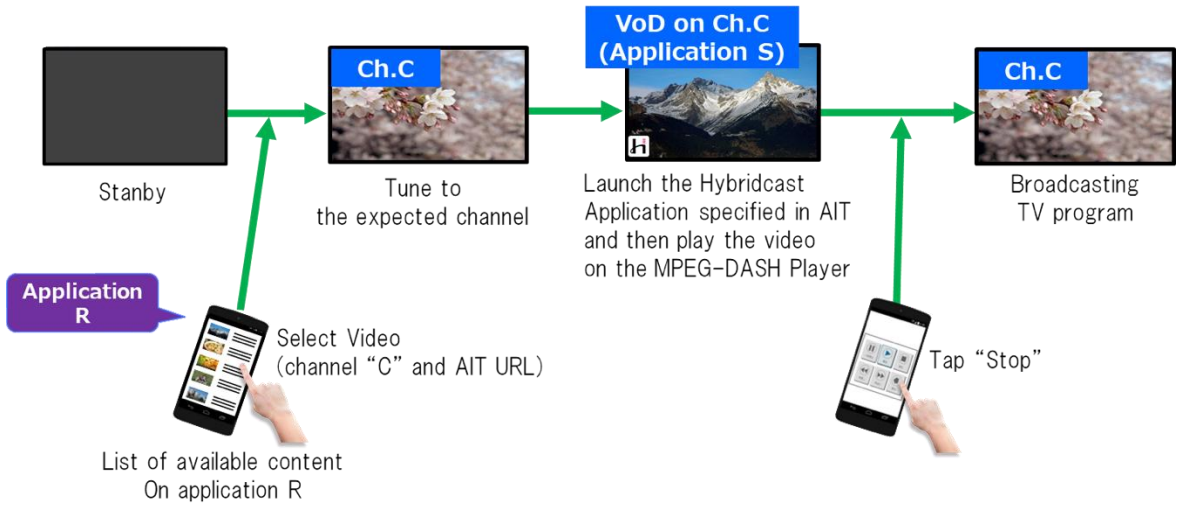
5) Application S sends Application R the URL of the application to provide a user interface for controlling on-demand playing.

6) Application R receives the URL from the receiver and presents a user interface for controlling on-demand playing.

7) When the user makes a request for stopping on-demand playing via the user interface, Application S stops on-demand playing and presents the programme aired by Broadcaster C.

FIGURE 1-34

**An example of on-demand services on the receiver controlled by an application on the companion device**




## 16 Multi-resolution video services using Hybridcast

TV monitors capable of displaying high-resolution video signals beyond HDTV have been widely spread and programmes of 4K UHDTV have already been provided for satellite broadcasting and video streaming services in Japan.

The Hybridcast system allows multi-resolution video services in which an HDTV programme via digital terrestrial broadcasting and an UHDTV programme via broadband are provided simultaneously. It can also provide VOD catch-up services. Operational guidelines for such services are available as IPTVFJ STD-0013 (in Japanese).

### 16.1 MPEG-DASH player for Hybridcast

MPEG-DASH video player named DashNX, which is a JavaScript library called DashNX.js and supports Hybridcast, has been developed. DashNX is designed to run on HTML5 and require small CPU processes and small memory usage for easy implementation in receivers including smart phones and PCs. It uses MSE (Media Source Extension) API for playback and supports DRM using EME (Encrypted Media Extensions) API, both of which are supported by HTML5.

### 16.2 Use cases of Multi-resolution video services using Hybridcast

#### 1) UHDTV simultaneous streaming with terrestrial digital broadcasting using Hybridcast

Fuji Television, a commercial broadcaster in Japan, provided a test service of simultaneous streaming of a 4K UHDTV with HDTV terrestrial digital broadcasting in October 2017. A 4K HDR version and a 4K SDR version of a programme were delivered via broadband using Hybridcast while an HDTV version was broadcast for terrestrial digital broadcasting.

Figures 1-35 and 1-36 show the system configuration and the receivers showing the programme, respectively. When the programme starts, a receiver downloads the start-up information file (XML-AIT: XML-Application Information Table) for a Hybridcast application from the URL multiplexed on a broadcast signal. The receiver activates the Hybridcast browser and opens the HTML5 application at the URL specified in XML-AIT. A JavaScript program determines whether the TV receiver supports the multi-period MPEG-DASH system. For a compatible receiver, a switching button is shown at the top on the screen so that the viewer can switch to the UHDTV version from the HDTV. The HTML5 application imports DashNX library and commands an API prepared

by DashNX to play a MPD (Media Presentation Description) file, a manifest file of MPEG-DASH video. DashNX accepts and interprets MPD, acquires video segment files and audio segment files and plays those files using the MSE API of the browser. Concerning video segment files, bitrate is adapted to the state of the broadband network. Bitrate of H.265|HEVC codec video segment files of 3 Mbit/s, 6 Mbit/s, 12 Mbit/s, and 24 Mbit/s were used in the test service. While playing the files, Stream Event multiplexed on broadcast signals are used to synchronize the UHDTV programme via broadband with the HDTV programme via broadcasting. When the programme ends, the screen automatically changes back to the HDTV broadcast.

FIGURE 1-35

**System configuration of multi-resolution video service using Hybridcast**

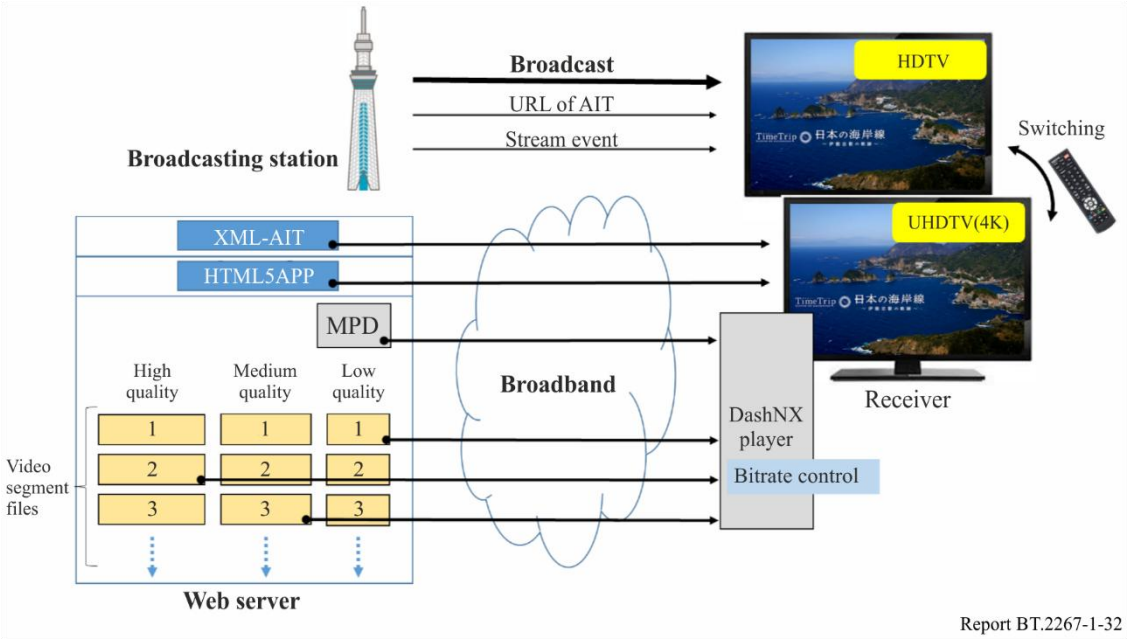
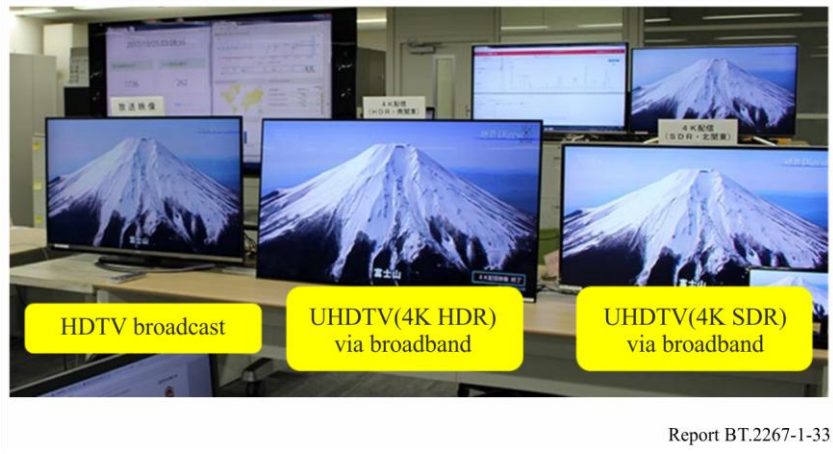



FIGURE 1-36

**Receivers with UHDTV simultaneous streaming using Hybridcast**




The main programme content and advertisements can be streamed separately from the cloud server. The multi-period MPEG-DASH system makes it possible to deliver them seamlessly. Addressable advertisements depending on viewers' local regions can also be used.

The system is equipped with a function that can automatically switch the view to the broadcasting channel when emergency is signalled by broadcasting so that the audience would feel safe and secure while watching UHDTV streaming.

#### 2) UHDTV live streaming and on-demand services

NHK provided trial 4K UHDTV streaming services and on-demand services using Hybridcast on the occasion of a global sports event in February 2018 in conjunction with the UHDTV satellite broadcasting, in which 8K and 4K HDR/SDR UHDTV programmes are provided. Some 4K streaming contents were simulcast of 8K/4K broadcasting and others were those unavailable for broadcasting.

## 17 Broadcast-based personalized video services using Hybridcast

### 17.1 Background

The Hybridcast system can provide personalized video services over terrestrial broadcasting. One example of a service leveraging Hybridcast functionality is the delivery of content that dynamically switches between the main programme and advertisements based on viewer characteristics.

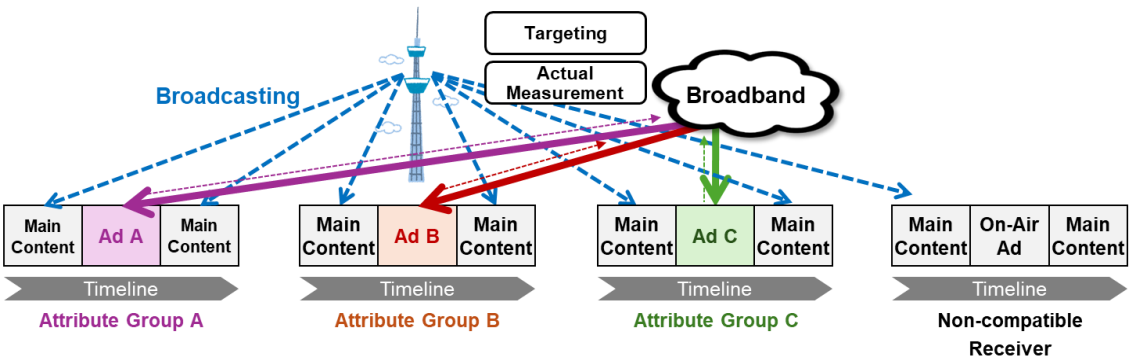
### 17.2 Use case

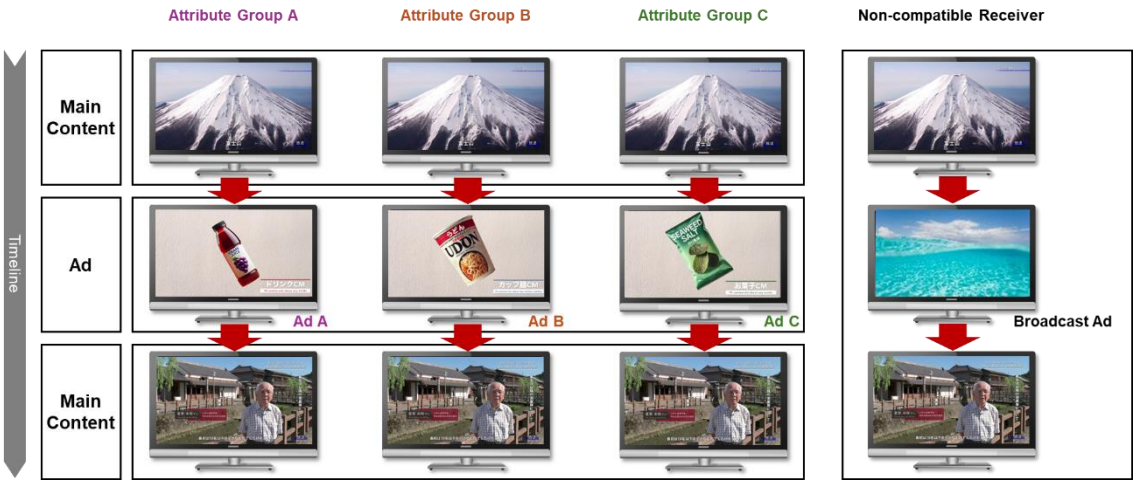
Fuji Television, a commercial broadcaster in Japan, launched an addressable television advertising service that delivers individualized advertisements to each receiver during current television broadcasting on 20 June 2025.

Technology has been developed that enables television receivers to process “digital advertising” while viewers are watching broadcast programmes. “Digital advertising” is a general term for advertising delivered over the Internet and is implemented in accordance with international digital advertising standards by the Interactive Advertising Bureau (IAB). While the main broadcast content is displayed as usual, the system switches to digital advertisements when the programme enters a Dynamic Advertising Substitution (DAS) slot. After the DAS slot ends, broadcast content resumes. This is the first addressable television advertising with DAS functionality using ISDB-T, Japan’s current broadcasting system. A conceptual diagram of broadband-based personalized video services is shown in Fig. 1-37. Receivers display different types of advertisements delivered via the Internet that correspond to any user attributes information stored locally.

FIGURE 1-37

**Addressable television advertising service**





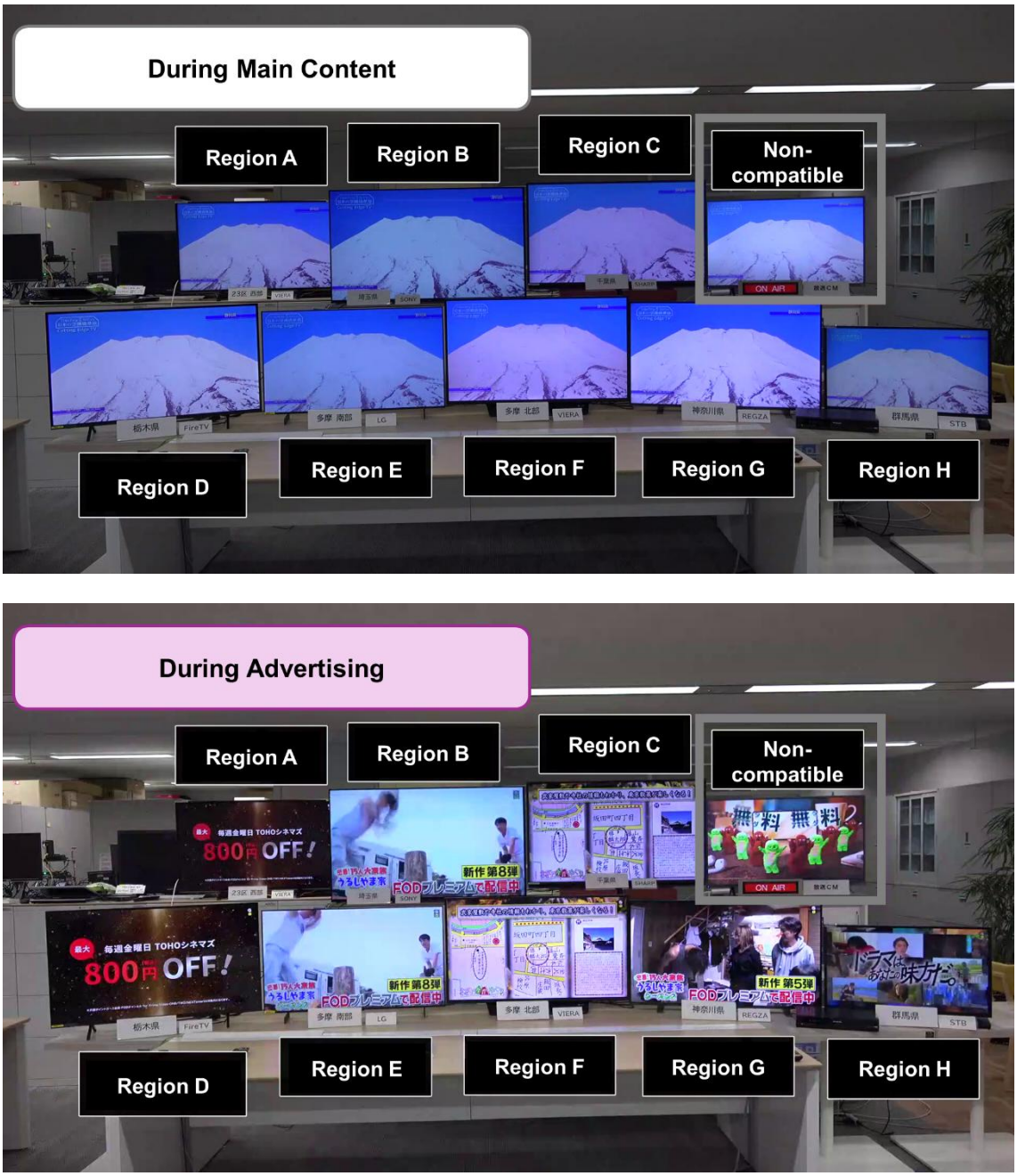

In this case, the broadcast area is divided into multiple regions and different advertisements are displayed based on the postal code configured in the receiver. The advertisements are selected and presented solely based on the postal code information, without transmitting any unique user identifiers or viewing history to the advertising servers ("ad servers") that distribute advertisements over the Internet. In different use cases, attribute information other than region may be used. Figure 1-38 shows receivers receiving addressable television advertisements. Each receiver was set to a different region's postal code and displayed a different advertisement. The receiver in the top right corner was a non-compatible model and displayed the broadcast advertisement.

In addition to displaying different advertisements, this approach also makes it possible to present different main programmes. This technology operates on receivers compatible with the "HC Video" specifications defined in IPTV Standard IPTVFJ STD-0013 "Hybridcast Operational Guideline." This standard specifies the operational guidelines for broadcast-oriented managed applications in Section 6 "Operational Guideline for Hybridcast Applications" and the implementation method for VOD services in Section 9 "Specifications for VOD operation with a video element". This use case is realized based on these specifications.

The switching between broadcast video and broadband-delivered video employs the same mechanism as described in § 16 "Multi-resolution video services using Hybridcast" of Annex 2. A broadcast-oriented managed application as defined in IPTVFJ STD-0013 is started and terminated at specified timings by control signals embedded in the broadcast signal. This application controls content presentation through Hybridcast HTML documents, in which an "object" element is used to display broadcast video and a "video" element is used to display broadband-delivered video. Switching between broadcast video and broadband-delivered video is achieved by controlling these elements, regardless of the video transport method. Synchronization of the playback timing between broadcast video and broadband-delivered video is achieved by transmitting relative timing information within the programme using control signals referred to as "EventMessage", which are transmitted within the transport stream.

FIGURE 1-38

**Receivers receiving addressable television advertisements**



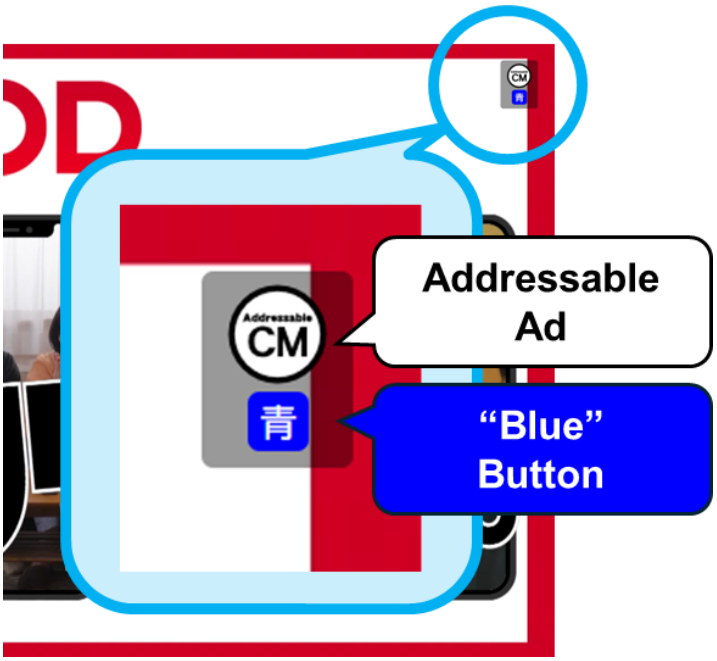

The service incorporates technologies designed to minimize any degradation in image quality during period of poor Internet connectivity by buffering. When interfacing with ad servers, the system employs technologies that are compatible with international digital advertising standards. Furthermore, the service can be implemented without modifying existing Hybridcast-compatible broadcasting equipment, allowing current terrestrial broadcasting services to continue uninterrupted. This has been achieved by integrating global digital advertising standard technologies into the Japanese broadcasting standard, ISDB-T.

In recent years, “fake advertising” has become a significant concern with the digital advertising ecosystem. To address this issue, countermeasure technologies for fake video advertising have been implemented. These technologies enable the verification of advertisement authenticity, including advertiser information and content provenance, utilizing cryptographic technologies that allow

viewers to confirm the legitimacy of advertisements. This approach ensures that viewers can trust advertising in a manner compatible to traditional broadcast advertising, while also preserving the confidence of advertisers. Figure 1-39 shows the presentation of content provenance information for countering fake video advertising.

FIGURE 1-39

**Presentation of content provenance information for countering fake video advertising**




a) Icon




b) Content provenance information

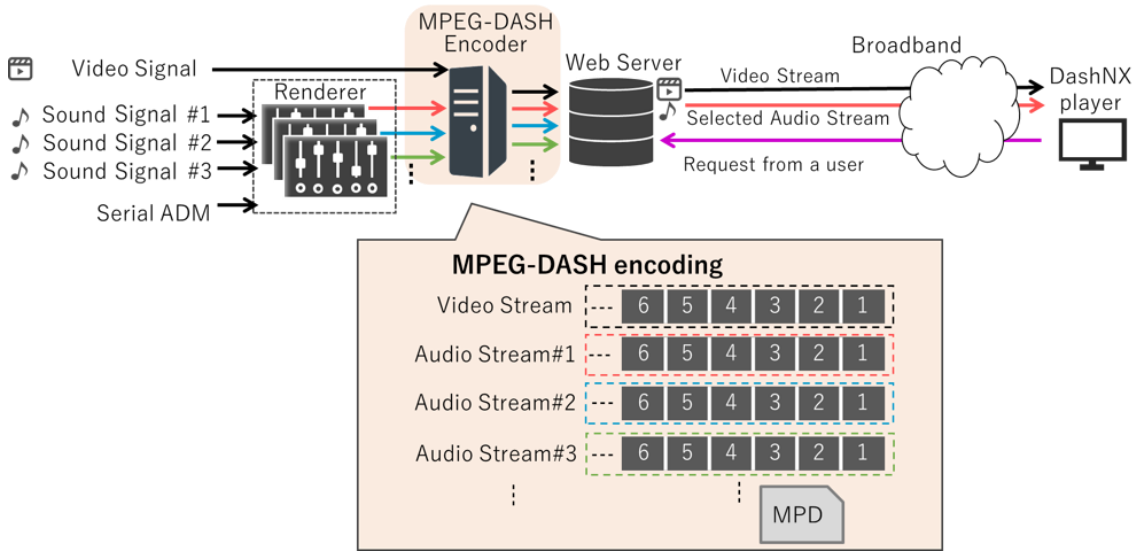
## 18 Object-based sound services using Hybridcast

Object-based sound systems can provide receivers with multiple audio signals associated with metadata specified in Recommendation ITU-R BS.2125. The receiver renders various sounds according to the metadata and listener's loudspeaker layout and preferences.

MPEG-DASH described in § 16 can also be used to provide object-based sound services. Figure 1-40 shows the overview of the system for distributing object-based sound services on Hybridcast. A live audio programme consists of one video stream and some audio streams. Different audio streams are produced by a renderer using the Serial ADM (S-ADM) with different service parameters (e.g. multilingual, dialogue enhancement, different mixing balances, and different loudspeaker layouts). The MPEG-DASH encoder produces video and audio streams with an MPD file, which is a manifest file of MPEG-DASH video to be stored in a Web server for Hybridcast. When a user requests one of the audio streams from the receiver, the Web server transmits the selected audio stream. In this way, the user can select a preferred audio stream seamlessly without adverse effects on the video stream.

FIGURE 1-40

**System example for distributing object-based sound services on Hybridcast**

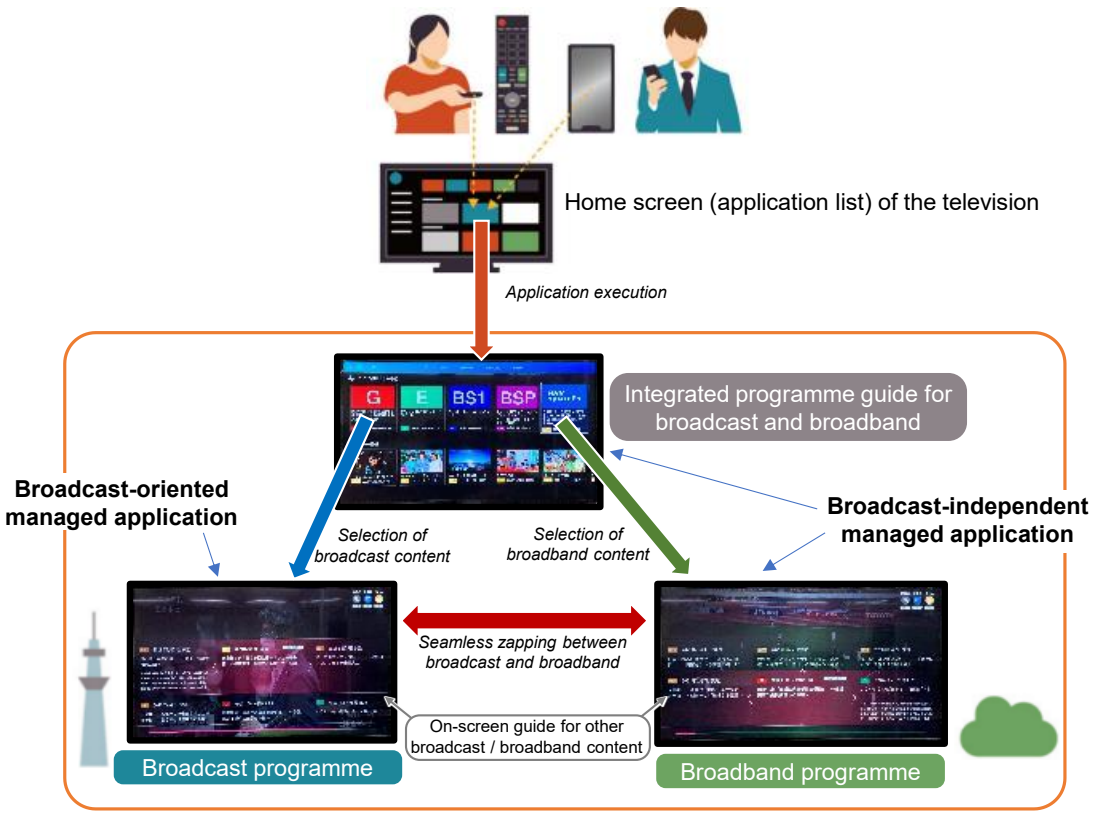



## 19 Receiver system using broadcast-oriented managed applications and broadcast-independent managed applications

With broadcast-oriented managed applications and broadcast-independent managed applications, seamless switching between broadcasting and broadband is possible. Figure 1-41 shows a navigation path for viewing broadcast and broadband content using a broadcast-oriented application and a broadcast-independent managed application. The broadcast-oriented managed application is associated with a specific broadcast channel and is used to present broadcast programmes with an on-screen guide for other broadcast/broadband content. The broadcast-independent managed application runs as a stand-alone application and is used as a broadcast-broadband integrated programme guide and for presenting content delivered via the broadband network.

FIGURE 1-41

**Navigation of broadcast and broadband content and corresponding application types on experimental receiver**




# Annex 3

# Integrated broadcast-broadband system based on enhancement of data broadcasting

## 1 Introduction

Since the launch of digital broadcasting services in Japan in 2000, interactive data broadcasting has been made possible by the use of Broadcast Markup Language (BML). BML is defined in ARIB STD-B24 and Recommendation ITU-R BT.1699, and is designed to provide bidirectional communication capability by using a return channel. However, in the early stages, the return channel was 2 400 bit/s analogue telephone line. It can be used to send a short text message from a receiver to a broadcaster, but it is impractical to use it as a forward channel. As the number of Web users increased rapidly, the broadband environment also made significant progress. In 2004, an Internet-based interactive data broadcasting service was launched for digital TVs equipped with an Ethernet port. The Internet-based interactive data broadcasting enables to combine interactive content over broadcast and additional interactive content over the Internet. One-Seg, which is aimed at handheld receivers and cell phones began in 2006, also offers interactive services that effectively link broadcast and telecommunications.

Recently, broadband network has reached more than 70% penetration in Japan and FTTH is the most major access method for broadband connection. This means that broadband networks can be used as

a practical channel to deliver audiovisual content. By using this infrastructure, some broadcasters launched a broadband-based video-on-demand service for digital TVs in 2009, and which is under the control of interactive content.

All the features mentioned above are achieved by the combination of broadcast and broadband networks, that is, hybrid broadcast services, and have been achieved by enhancement of BML functionalities. Hybrid broadcasting services using enhanced BML are broadcast-centric as they are always activated from normal data broadcasting. Additional functionalities of BML for hybrid services are described below.

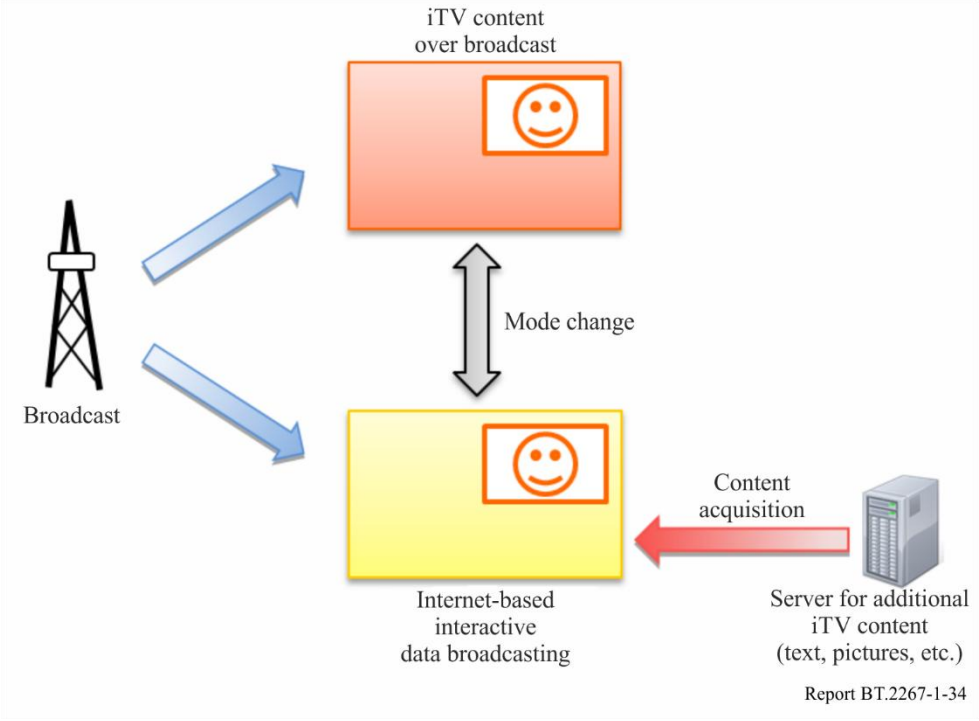
## 2 Internet-based interactive data broadcasting

Interactive data broadcasting or interactive TV (iTV) content is delivered over broadcast channel. Because broadcast channel is unidirectional, all the iTV content elements have to be delivered together simultaneously and a receiver selects required elements from the delivered elements according to end-users' instruction for presentation. Content elements to deliver is sometimes limited by the available transmission bandwidth. However, if content elements are delivered over broadband channel, such limitation can be overcome by use of server-client data transfer like Web.

BML has ability to interact with web servers. The mechanism is quite simple; only to specify a web server instead of broadcast carousel. Obtained content elements from the web servers are parts of broadcast service and thus simultaneous presentation of broadcast AV content is allowed. Figure 1-42 shows the behaviour of a receiver in this case.

FIGURE 1-42

**Internet-based interactive data broadcast**




However, security mechanism for Internet-based data broadcasting is introduced into the receivers to maintain content integrity. Allowable sites or domains for each broadcast channel are presented to a receiver by manufacturers. If iTV content instructs to get content elements or BML documents from the sites outside of allowable areas, a receiver terminates the presentation of broadcast AV content.

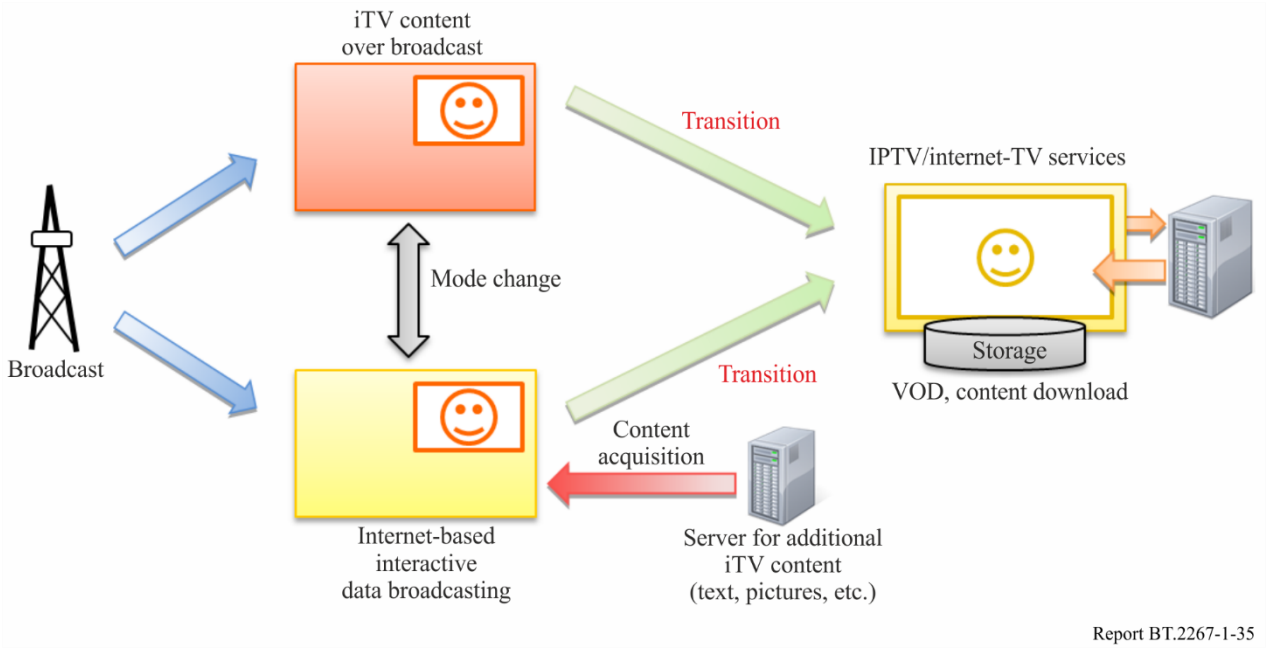
## 3 Switching to IPTV/Internet-TV[7] services

Interactive data broadcasting can be used as a gateway to IPTV/Internet-TV services including video on demand (VOD). To enjoy VOD services, viewers need to select content to play and command equipment to play it. Switching to VOD portal site from interactive data broadcast content is the simplest way to follow needed actions. This is one of the suitable ways to take the viewers to IPTV/Internet-TV services where the broadcasters offer their specific services.

A function added to script language of BML starts another browser in a different language such as HTML. The function instructs initial URL to the other browser and terminates the BML browser where the script runs. The function can be applied in both normal data broadcast and Internet-based interactive data broadcast described in the previous clause. Figure 1-43 shows transition paths to IPTV/Internet-TV services by using this function.

FIGURE 1-43

**Switch to Internet-TV services**




The function also “suggests” URL of the original interactive content to the other browser to return to broadcast-based services. However, the return to the original content is not guaranteed. Similar to start the other browser, the BML browser has to be restarted and the BML browser may not accept “suggested” URL for return. (Because Japanese data broadcast is operated in a way that an entry document is always the same, the BML browser does not need to provide ability to receive URL as an invocation parameter.) Another way to go back to the broadcast service is to tune the channel again.

## 4 Direct access to IPTV/Internet-TV services

In contrast with switching to IPTV/Internet-TV services described in the previous clause, this functionality employs IPTV/Internet-TV services as a part of interactive data broadcast services. This functionality is intended to incorporate VOD and content download offered in IPTV/Internet-TV

[7] The term “Internet-TV” is used in this Report to indicate interactive services providing delivery of video/audio/metadata/multimedia information over unmanaged Internet networks.

services from both normal data broadcast content and Internet-based interactive data broadcast content.

For this functionality, other additional functions of script language of BML are added. The functions are to call receiver-resident VOD playback and content download functions. Figure 1-44 shows this functionality.

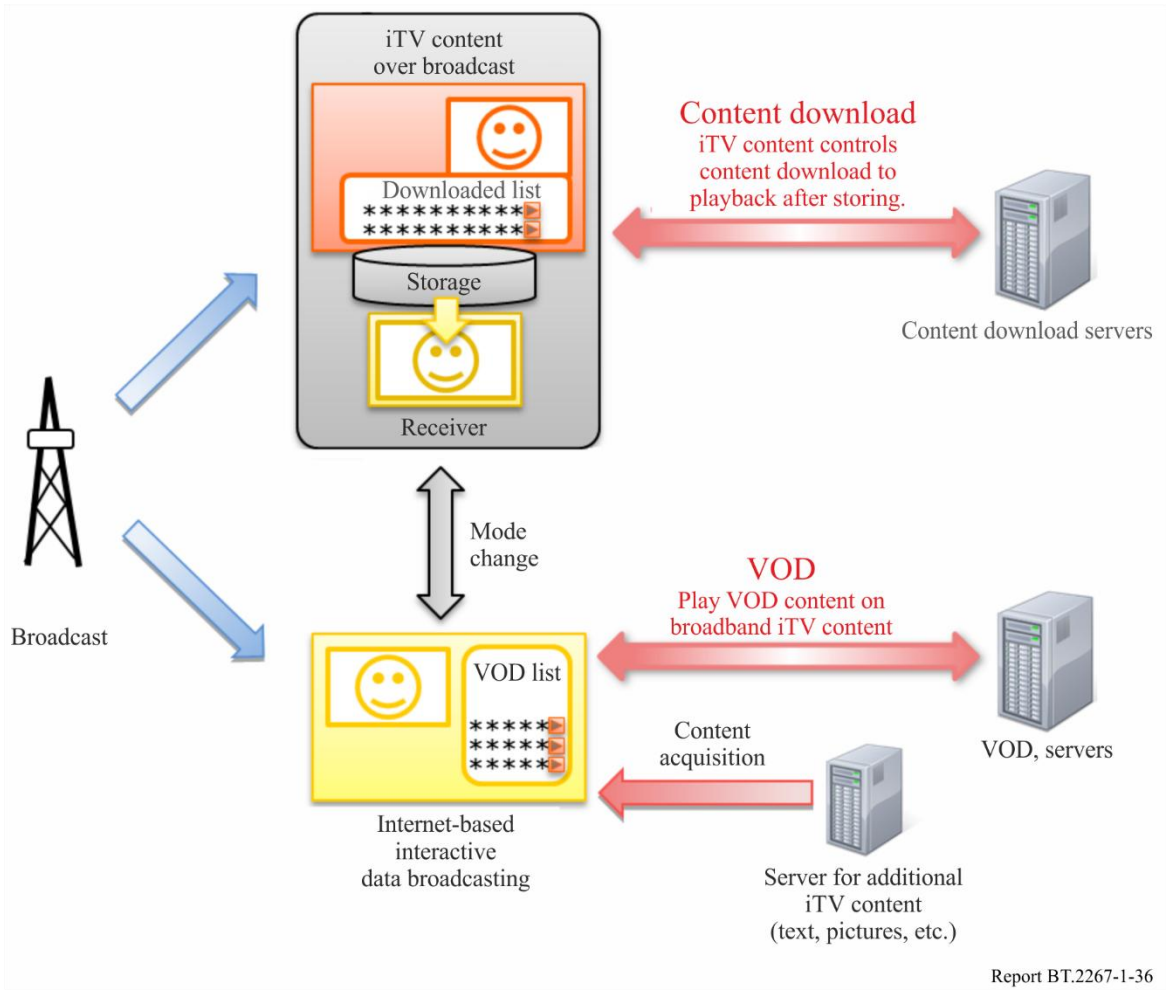
When VOD related function is called, a receiver-resident VOD player starts in response to the function call. The VOD player obtains streaming signal for specified VOD content. When the VOD player finishes its presentation, the same page of BML content as the calling page is presented. By using this function, keeping user interface, offer of trailer video clips and catch-up service are available.

In normal interactive data broadcast content, while presenting VOD content, data broadcast content may be changed according to the progress of broadcast programme. To avoid the loss of the return page of BML document, call of the VOD related function is limited from Internet-based data broadcasting content only.

Similarly, when content download function is called, a receiver-resident content downloader starts. The content downloader runs in background and stores specified content in a receiver built-in storage device. Because this function returns immediately after start of the content downloader, this function can be called from normal interactive data broadcast content.

FIGURE 1-44

**Direct access to IPTV/Internet-TV services**



### 5 Compatibility consideration

A receiver equipped with enhanced BML functionalities also provides capability of probing functions. By using these functions, the interactive data broadcast content can be authored to maintain compatibility with legacy models or receivers. That is, the interactive data broadcast content can call enhanced functions only on the capable receivers.

### 6 Future development

BML is designed specifically for interactive data broadcast. To get more flexibility and compatibility with the latest web technologies, development and standardization of a new system based on HTML5 is underway. The new system will be introduced into the market in two steps. The first step will provide rather simple functionalities similar to the interactive data broadcasting including functionalities described above. The second step will provide more complex and advanced functionalities such as precise cross-stream synchronization or third-party made applications. Even in the first step, capability of dynamic JavaScript libraries, flexibility of presentation and compatibility with existing web will allow wider range of services.

## Annex 4

## Integrated broadcast-broadband system based on the Ginga middleware

### 1 Introduction

Ginga is divided into four main integrated subsystems, which allow for the development of applications following different programming paradigms. Those subsystems are called Ginga-CC (Common Core), Ginga-NCL (for declarative NCL applications), Ginga-HTML5 (for declarative HTML5 applications) and Ginga-J (for imperative Java applications). Currently only the first two subsystems, Ginga-CC and Ginga-NCL are mandatory for all Ginga implementations. Ginga-CC delivers applications with their associated media content – which can come from broadcast signals or be delivered by broadband IP services – to Ginga-NCL, Ginga-HTML5 or Ginga-J, depending on the type of application. Ginga-CC includes a remote API that allows for the access and certain degree of control over the broadcast services, based on the REST architectural style. Ginga-CC WebServices (ABNT NBR 15606-11) provides this remote API to broadcaster-authorized Ginga and non-Ginga applications. In this way, any application running on devices in the home network (TV, SmartTV, SmartPhone, etc.) may be authorized to be part of the IBB experience. Ginga-NCL is the Presentation Environment of Ginga, the logical subsystem responsible for running NCL applications.

NCL is the declarative language of Ginga. Its characteristics make it a sound declarative solution for IBB services: the language flexibility; its reuse facility; multi-device support; presentation and application content adaptability; API for building and modifying applications on-the-fly; and, mainly, its intrinsic ability for easily defining spatiotemporal synchronization among media assets (including viewer interactions). For particular procedural needs, e.g. when more complex dynamic content generation is required, NCL provides support to the Lua scripting language. Lua is a powerful, fast, lightweight, embeddable scripting language.

Ginga has since evolved within the Brazilian TV 3.0 system (also known as DTV+), whose application coding is specified in ABNT NBR 25608. In this evolution, Ginga has become an

application-oriented television broadcasting platform, as described in Report ITU-R BT.2568, which presents TV 3.0 as an example of such a system in its Section 5.

## 2 Ginga position

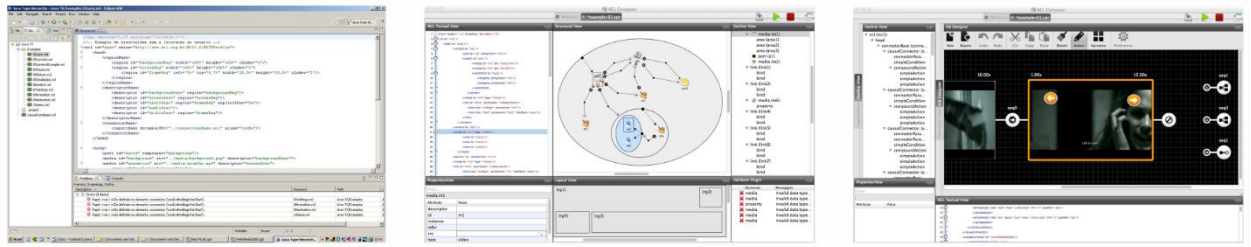
Ginga was developed in 2006 and first standardized by ABNT on November 30, 2007 (see References). The first commercial Ginga-NCL implementations appear in 2008. In 2009, Ginga-NCL and NCL 3.0 became part of Recommendation ITU-T H.761 for IPTV services and Recommendation ITU-R BT.1699. Many countries, particularly in South America region, have adopted Ginga as the middleware of their terrestrial DTV standards, based on the Brazilian-Japanese ISDB-T Standard. The Ginga specifications have continuously evolved ever since through the introduction of new receiver profiles. The first profile introduced is known as profile A (FSA_09 for full-seg or OSA_09 for one-seg) and the latest profile is known as profile D (FSD_09 for full-seg or OSD_09 for one-seg).

Consumer equipment manufacturers offer a large number of models of Ginga enabled TV sets, set-top boxes and mobile (smart) phones. There are open-source implementations for Linux, Windows, MAC OS, and Android platforms which can be embedded in desktop computers, tablets, smart phones, etc. Some consumer equipment manufacturers offer these open-source implementations in their products.

Several textual and graphical authoring tools were developed to help application designers to create new NCL applications. Figure 1-45 exemplifies some of them.

FIGURE 1-45

**Some NCL authoring tools**



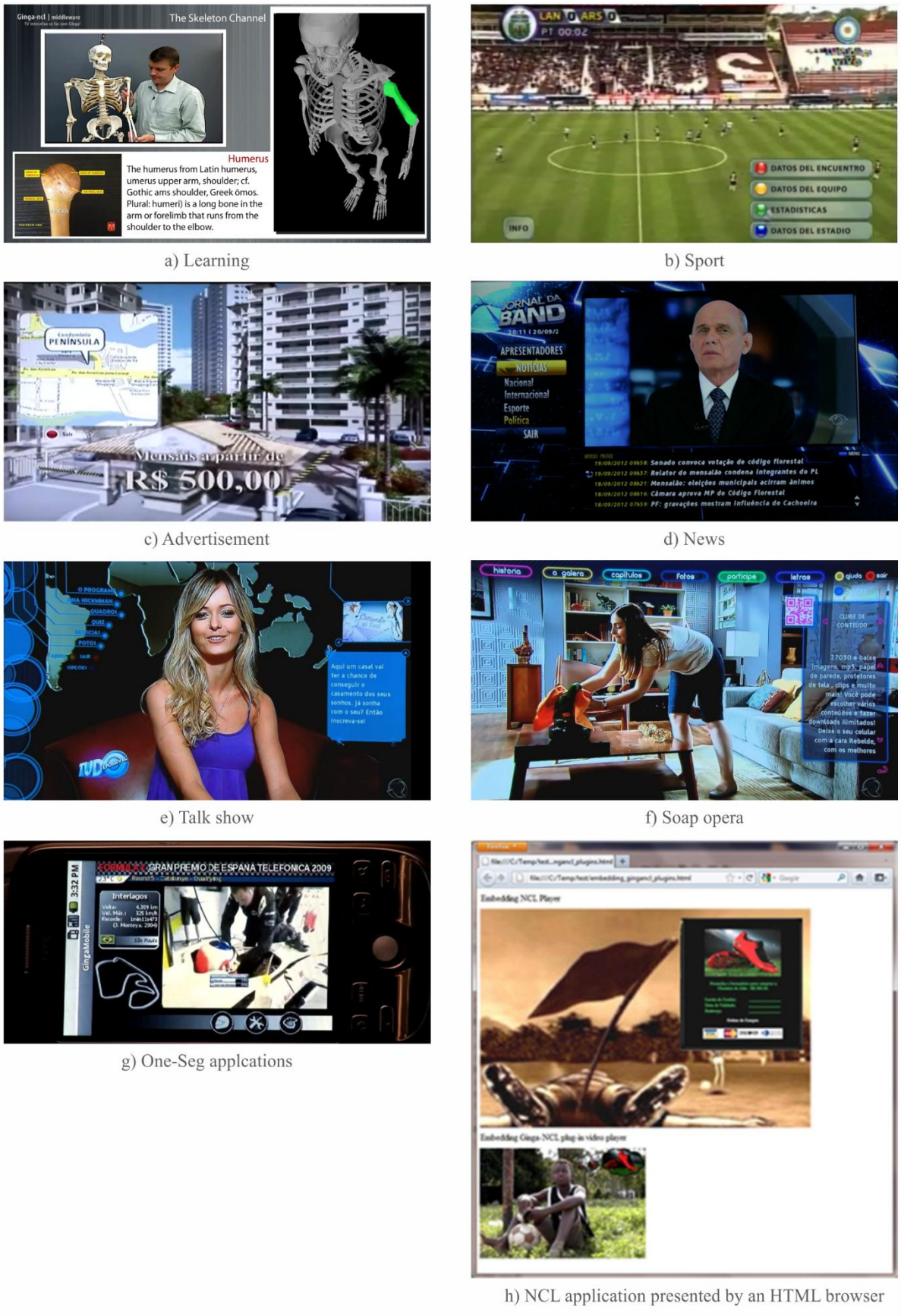
a) NCL eclipse b) NCL composer a) ISB designer

Report BT.2267-1-37

Up to now, a high number of applications have been developed for NCL. They come mainly from broadcasters in Brazil and Argentina, but also come from other countries that have adopted Ginga. A number of applications are also available in the Web. Several Web browsers are able to play NCL applications natively or using NCL plug-ins.

FIGURE 1-46

**Some Ginga applications**




a) Learning

b) Sport

c) Advertisement

d) News

e) Talk show

f) Soap opera

g) One-Seg applcations

h) NCL application presented by an HTML browser

Report BT.2267-1-38

As mentioned previously, Ginga middleware specifications comprise different profiles, with different application environments and with different media type supports. Table 1-2 summarizes the differences among Ginga receiver profiles.

TABLE 1-2

**Ginga receiver profiles**

| **Category** | **Media type** | **Ginga receiver profile A (full-seg: FSA_09, one-seg: OSA_09)** | **Ginga receiver profile B (full-seg: FSB_09, one-seg: OSB_09)** | **Ginga receiver profile C (full-seg: FSC_09, one-seg: OSC_09)** | **Ginga receiver profile D (full-seg: FSD_09, one-seg: OSD_09)** |
|---|---|---|---|---|---|
| Application environment | Ginga-NCL (see Note 1) | Mandatory | Mandatory | Mandatory | Mandatory |
| | Ginga-J (see Note 2) | one-seg: not supported full-seg: Optional | one-seg: not supported full-seg: Optional | one-seg: not supported full-seg: Optional | Not supported |
| | Ginga-HTML5 | Not supported | Not supported | Not supported | Mandatory |
| | Ginga CC WebServices | Not supported | Not supported | Not supported | Mandatory |
| Images | PNG | Mandatory | Mandatory | Mandatory | Mandatory |
| | GIF (see Note 3) | one-seg: Mandatory full-seg: not supported | one-seg: Mandatory full-seg: not supported | one-seg: Mandatory full-seg: not supported | Mandatory |
| | JPEG | Mandatory | Mandatory | Mandatory | Mandatory |
| | MNG | one-seg: not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory | Not supported |
| Text | HTML (see Note 4) | Mandatory | Mandatory | Mandatory | Mandatory |
| | XML | Mandatory | Mandatory | Mandatory | Mandatory |
| | CSS (see Note 4) | Mandatory | Mandatory | Mandatory | Mandatory |
| | TXT | Mandatory | Mandatory | Mandatory | Mandatory |
| Resident fonts | Tiresias | one-seg: not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory |
| | Verdana | one-seg: Mandatory full-seg: Not Supported | one-seg: Mandatory full-seg: Not Supported | one-seg: Mandatory full-seg: Not Supported | one-seg: Mandatory full-seg: Not Supported |
| Downloadable fonts | OFF | one-seg: not supported full-seg: Mandatory | one-seg not supported full-seg: Mandatory | one-seg: not supported full-seg: Mandatory | Mandatory |
| | WOFF | Not supported | Not supported | Not supported | Mandatory |
| Text coding | UTF-8 | Mandatory | Mandatory | Mandatory | Mandatory |
| | ISO/IEC 8859-1 | Mandatory | Mandatory | Mandatory | Mandatory |

TABLE 1-2 (*end*)

| Category | Media type | Ginga receiver profile A (full-seg: FSA_09, one-seg: OSA_09) | Ginga receiver profile B (full-seg: FSB_09, one-seg: OSB_09) | Ginga receiver profile C (full-seg: FSC_09, one-seg: OSC_09) | Ginga receiver profile D (full-seg: FSD_09, one-seg: OSD_09) |
|---|---|---|---|---|---|
| Video | MPEG-1 | Not supported | Mandatory | Mandatory | Not supported |
| | H.264 (see Note 3) | Not supported | Not Supported | Mandatory | Mandatory |
| | H.265 | Not supported | Not supported | Not supported | Mandatory |
| Audio | MPEG-4 AAC (see Note 3) | Mandatory | Mandatory | Mandatory | Mandatory |
| | MPEG-1 (Layers 1&2) | Mandatory | Mandatory | Mandatory | Not supported |
| | AC-3 | Not supported | Not supported | Not supported | Mandatory |
| | E-AC-3 | Not supported | Not supported | Not supported | Mandatory |
| | AC-4 | Not supported | Not supported | Not supported | Optional |
| | MPEG-H 3D Audio | Not supported | Not supported | Not supported | Optional |
| Subtitles | IMSC1 TTML | Not supported | Not supported | Not supported | At least one of the subtitle formats must be supported |
| | WebVTT | Not supported | Not supported | Not supported | |
| Multimedia | MPEG-4 container | Not supported | Not supported | Mandatory | Mandatory |
| Adaptive streaming | MPEG-DASH | Not supported | Not supported | Not supported | At least one of the adaptive streaming formats must be supported |
| | HLS | Not supported | Not supported | Not supported | |
| DRM | Compatible with MPEG-CENC | Not supported | Not supported | Not supported | Mandatory |

Note 1: Legacy implementations of the one-seg profiles OSA_09, OSB_09 and OSC_09 can use ABNT NBR 15606-5 instead of ABNT NBR 15606-2. However, new implementations of these profiles must use ABNT NBR 15606-2. The one-seg profile OSD_09 must use ABNT NBR 15606-2. Some functionalities of NCL and Ginga-NCL are specific to some profiles, as defined in ABNT NBR 15606-1, Annex B.

Note 2: When Ginga-J is implemented, the bridge between Ginga-NCL and Ginga-J must be implemented.

Note 3: Some coding restrictions vary among different profiles, as defined in ABNT NBR 15606-1, Annex A.

Note 4: The restrictions for HTML and CSS formats in Ginga receiver profiles A, B and C are defined in ABNT NBR 15606-2. The restrictions for HTML and CSS formats in Ginga receiver profile D are defined in ABNT NBR 15606-10.

Ginga receiver profile D includes several new monomedia/multimedia formats (mandatory or optional), such as: H.265/HEVC video, E-AC-3, AC-4 and MPEG-H 3D Audio, and IMSC1 TTML or WebVTT subtitles. Supported video resolutions include 4K UHDTV and video encoding includes the possibility of using high dynamic range (HDR). It also includes support for adaptive streaming and DRM.

## 3 Ginga architecture

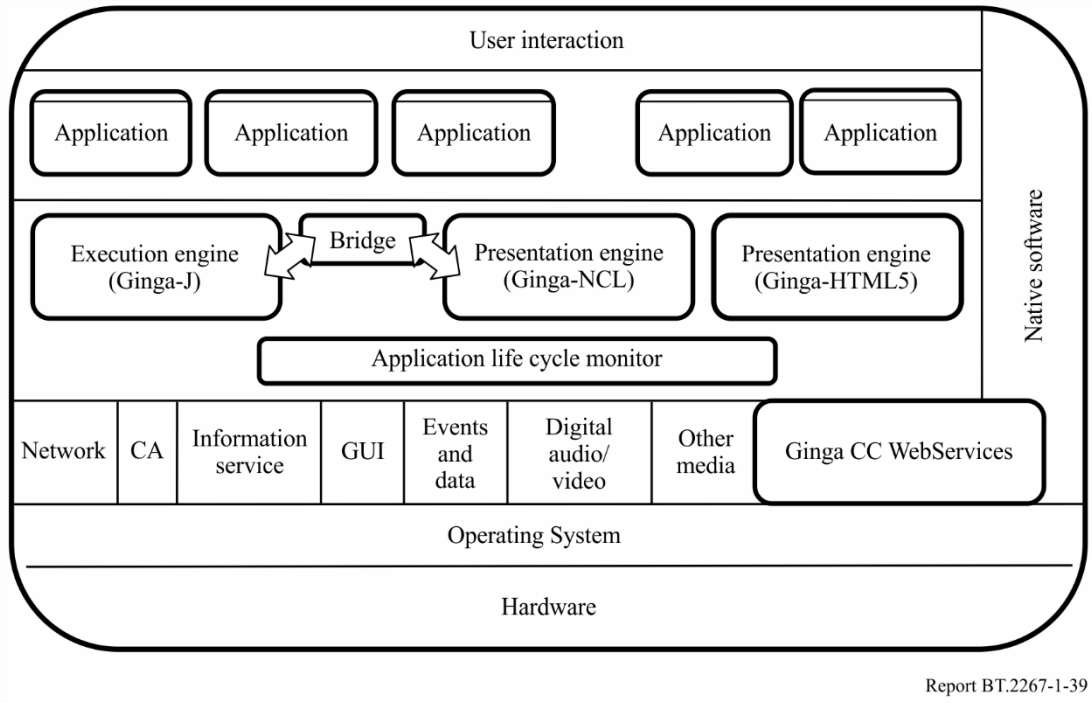
The Ginga architecture is depicted in Fig. 1-47. Ginga allows for some extensions, but they are optional. To be Ginga compliant, the Ginga-NCL subsystem is required. The overall architecture for Ginga interactive television should be in accordance with Recommendation ITU-T J.200 and it includes three application environments: the Ginga-J execution engine (optional in receiver profiles A to C, not applicable in receiver profile D) and the Ginga-NCL (mandatory in all interactive receiver profiles – A to D) and Ginga-HTML5 (mandatory in receiver profile D) presentation engines. Receivers implementing the Ginga-J execution engine must also implement the bridge between the Ginga-J and Ginga-NCL engines.

The Ginga Common Core (Ginga-CC) is composed of media players, of procedures to obtain contents that can be transported in diverse networks accessed by a receiver (see § 4), and of the conceptual display graphical model defined by the receiver platform. The Ginga Common Core is also responsible for gathering metadata information and providing this information to applications; for providing an API to communicate with DRM system; for managing context information (like user profiles and receiver profiles); and for supporting software version management (update) of Ginga's components.

Media player components serve application needs for decoding and presenting content types such as PNG, JPEG, MPEG and other formats, including those for content which contains declarative or imperative code, like HTML, Lua, etc. A generic media player API establishes the necessary communication between media player components and the Ginga-NCL subsystem. Thanks to this API, Ginga-NCL and the Ginga-CC are strongly coupled but independent subsystems. Ginga-CC may be substituted by other third part implementations, allowing Ginga-NCL to be integrated in other DTV middleware specifications, extending their functionalities with NCL facilities. Players that do not follow the generic API are required to use the services provided by Adapters.

FIGURE 1-47

**Ginga architecture**



The core of Ginga-NCL subsystem is the NCL Player. This component is in charge of receiving and controlling multimedia applications written in NCL. Applications are delivered to the NCL Player by the Ginga Common Core.

In Ginga-NCL, a declarative application can be generated or modified on the fly, using NCL editing commands (see § 6).

The NCL Player deals with NCL applications collected inside a data structure known as private base. A Private Base Manager component is in charge of receiving NCL editing commands and maintaining the NCL documents being presented (see § 6).

Ginga-NCL Presentation Engine supports multiple presentation devices through its Layout Manager module. This component is responsible for mapping all presentation regions (see § 5) defined in an NCL application to canvas on receiver's exhibition devices.

In particular, Ginga-NCL provides declarative support to IPTV specific services, such as VOD, datacasting, etc. Thus, a VoD service may, for example, play an NCL application besides the main audiovisual stream. Moreover, an IPTV service itself can be an NCL application.

Ginga-J is the procedural environment of Ginga, based on the Java language. Its stack includes the core API from Recommendation ITU-T J.202 and Recommendation ITU-R BT.1722 (specifically, CDC 1.1, FP 1.1, PBP 1.1.2 and JavaTV 1.1), the JavaDTV set of APIs as defined in ABNT NBR 15606-6, and a set of system-specific API defined in the ABNT NBR 15606-4. Ginga-J is currently optional for Ginga implementations.

Ginga receiver profile D includes a new presentation engine – Ginga-HTML5 – and a remote API server component, called Ginga CC WebServices.

The Ginga-HTML5 presentation engine defined in Ginga receiver profile D standardizes a base HTML5, CSS, and JavaScript profile to be supported by interactive DTV receivers, in accordance with ABNT NBR 15606-10.

The purpose of Ginga-HTML5 standard is to define an HTML5 profile that developers of interactive applications can use, based on current W3C standards. Ginga-HTML5 does not specify extensions to the mentioned technologies, but only a subset of interest for the development of DTV interactive applications.

Ginga-HTML5 maintains compatibility with the W3C-compliant sub-profile specified in the HbbTV 2.0.1 Specification.

## 4 Integrated broadcast and broadband delivery

Since its first standardized version in 2007, Ginga provides support to converged services by making use of broadcast and IP distribution path.

A DSM-CC Object Carousel is used as transport protocol for applications delivered within the DTV signal. Alternatively, applications can be retrieved through a broadband channel by using the HTTP protocol. The transport protocol mechanism is signalled in the AIT by using the transport protocol descriptor in accordance with ABNT NBR 15606-3.

Once the application is being executed in the receiver, it can use protocols such as HTTP, HTTPS, or more basic IP-based protocols such as TCP sockets and UDP, to communicate with servers or retrieve additional resources (code, images, video, audio, etc.) through a broadband channel.

Advanced application support in Ginga, including IBB (Integrated Broadcast and Broadband) applications, is provided by functional components and APIs defined throughout its subsystems, which include the Ginga Common-Core, Ginga-NCL, Ginga-HTML5 and Ginga-J. The current

specifications, standardized under the ABNT NBR 15606 series, fully address IBB requirements as specified in Recommendation ITU-R BT.2053.

NCL applications have a stricter separation between its content and its structure. NCL does not define itself any media content. Instead, it defines the glue that holds media objects together in multimedia presentations. An NCL document (NCL application specification) only defines how media objects are structured and related, in time and space. As a glue language, NCL does not restrict or prescribe the content types of its media objects. Which media objects are supported depend on the media players that are coupled in the NCL Presentation Environment. One of these players is the main video and main audio decoder/player, usually implemented using hardware resources in a DTV receiver. Therefore, the main video and the main audio of a service are treated like all other media objects that may be related using NCL.

Another media object that is required in a Ginga-NCL implementation is the HTML-based media object. Which HTML-based language will have support in an NCL formatter is an implementation choice, and, therefore, it will depend on which HTML browser will act as a media player integrated to the NCL presentation engine.

Still another media object that is required in a Ginga-NCL implementation is the declarative NCL media object, that is, a media object containing an NCL application. Therefore, NCL applications can be embedded in NCL parent applications, likewise HTML-based applications can be.

To extend the NCL declarative language basic model adding decision-making features that deserves the imperative paradigm, NCLua objects are part of the Ginga-NCL specification. Lua is the scripting language of NCL. NCLua media objects carry Lua code based on the standard NCLua API.

Each media object of NCL specifies in its *src* attribute the URI Scheme used to retrieve its content. Depending on the specified scheme, Ginga knows if it has to get the content from the broadcast signal or from the IP network. For example, if the URI is http://server_identifier/file_path/#fragment_identifier, the content must be retrieved from the IP network, if it is isdb://program_number.component_tag, it must be retrieved from the tuned channel signal.

The advent of new multimedia services, such as ultra-high-definition TV or free viewpoint TV, will put a very large bandwidth pressure on the broadcast networks. In order to offer these services while maintaining quality, enhancing broadcast services with broadband media may be one key point to low-cost deployment of new services. Integrated broadcast and broadband retrieval of audio-visual content in NCL allows for specifying URI schemes that locate the enhancement streams besides the usual URI scheme to retrieve the base stream in a broadcast signal.

In the case of Ginga-HTML5, IBB functionalities needed are provided by a remote API based on the REST architectural style, provided by ‘Ginga Common Core Web Services’ (ABNT NBR 15606-11). Therefore, Ginga-HTML5 applications access functions and data via HTTP requests and get the returned values mostly in JSON format. Ginga-NCL (ABNT NBR 15606-2) applications may also access this remote API, if needed, although most IBB functionalities can also be found in its NCLua API.

## 5 Secondary devices (second screens)

Since its first standardized version of 2007, NCL and Ginga-NCL provide declarative support for presenting DTV applications distributed on multiple devices, namely multiple device exhibition and multi-user control.

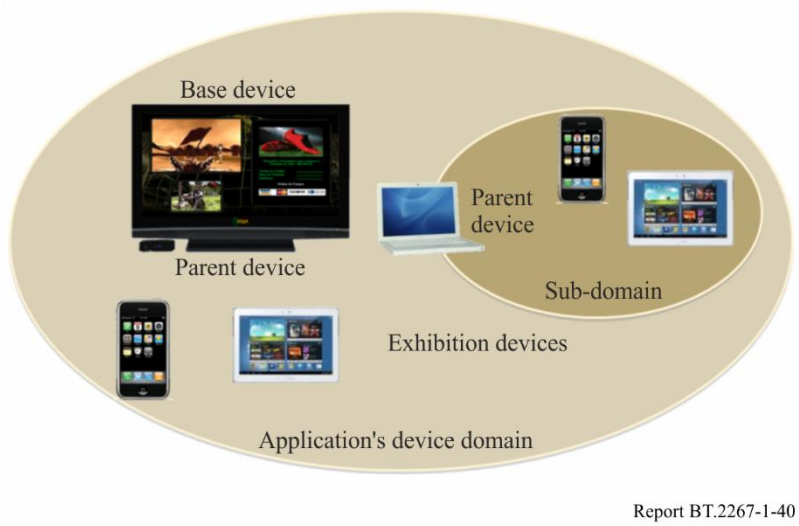
Presentation of media objects of NCL applications can be associated to devices using the abstraction called *device classes*. Secondary devices (child devices) are registered to classes controlled by a parent device.

NCL 3.0 has originally defined two types of device classes: the passive one, in which the same content is shown on all paired devices under the rendering of their parent device; and the active one, in which the content exhibition is rendered by each individual child device, thus allowing independent navigation and interaction.

The parent device can be the base device or any descendent secondary device. The parent device can delegate control to child devices registered in its active classes. From these secondary devices, it is possible to create new sub-domains (new classes under control of the new parent device), forming a hierarchical application domain, as shown in Fig. 1-48.

FIGURE 1-48

**Hierarchical device control model**




In defining a class, it is possible to define the maximum and minimum number of devices it has. This mechanism comes together with a method to identify each device in a class. For each media object, NCL allows for defining in which class of devices must be exhibited and in which region on the screens of all devices registered in the class.

Three classes are pre-defined in Ginga-NCL: passive Class 1, and active Classes 2 and 3. Devices registered in Class 1 can present pre-rendered encoded media streams received from a parent device in the hierarchical orchestration tree to which it is paired, and to navigate in such content as a group; that is, any navigational action is reflected in all paired devices (individual navigation is not allowed). Devices registered in Classes 2 and 3 can individually control the decoding and/or the execution of an NCL application (Class 2) or HTML application (Class 3) sent by the parent device, offering an individual (personalized) navigation on the content.

Figure 1-49 shows an NCL application exhibited in South Africa during the FIFA World Cup 2010, announcing the World Cup 2014. The base device was a TV set and several smart phones were used as secondary devices. Additional information about the previous world champions were presented on the secondary screens; in another moment, it was possible to buy tickets for the World Cup 2014; and finally a game was presented near the end of the video, in which a device could act as a penalty shooter and another as a goalkeeper. The result of the game was displayed in the TV set.

FIGURE 1-49

**Multi-screen application**



Report BT.2267-1-41

In NCL 3.0, dynamic class specifications are not supported. In addition, which protocols underlie the communication between the cooperating devices is left to proprietary implementations, although they shall follow the standardized multiple device API.

Ginga CC WebServices is a new component to be implemented in Ginga receiver profile D that provides APIs for access to broadcast information from any application (including Ginga or native applications), both from the receiver itself and from other devices present in the home environment according to ABNT NBR 15606-11.

Ginga CC WebServices provides a set of services implemented in the form of REST APIs, whose main objective is to complement the functionalities capable of promoting an integrated experience of the broadcast environment with the home and broadband environments.

As Ginga CC WebServices is implemented as a server, it has the following characteristics:

1) Its APIs are accessible from any underlying platform present in the interactive DTV receiver itself. Native Smart TV applications, Web-based applications and Ginga applications can access the available APIs;

2) Its APIs are also accessible to devices present in the home environment, regardless of having a corresponding interactive application running on the receiver, allowing, for example, to explore second screen (companion device) scenarios with a high-level interoperability and flexibility.

The set of extended functionalities provided by Ginga CC WebServices includes not only functions for accessing broadcast content, but also functions that standardize access to advanced features of the home environment (for example, general platform features, supported DRM schemes, advanced media playback capabilities). Another important feature of Ginga CC WebServices is the support for the use of deep links that target on applications installed on the interactive DTV receiver (native Smart TV applications, Web-based applications and Ginga applications) from any other application running on the interactive DTV receiver or on devices present in the home environment.

Ginga CC WebServices also includes features to ensure the safety and the privacy of the end user and the broadcaster control over its content. For any application not sent over-the-air by the broadcaster these conditions must be followed:

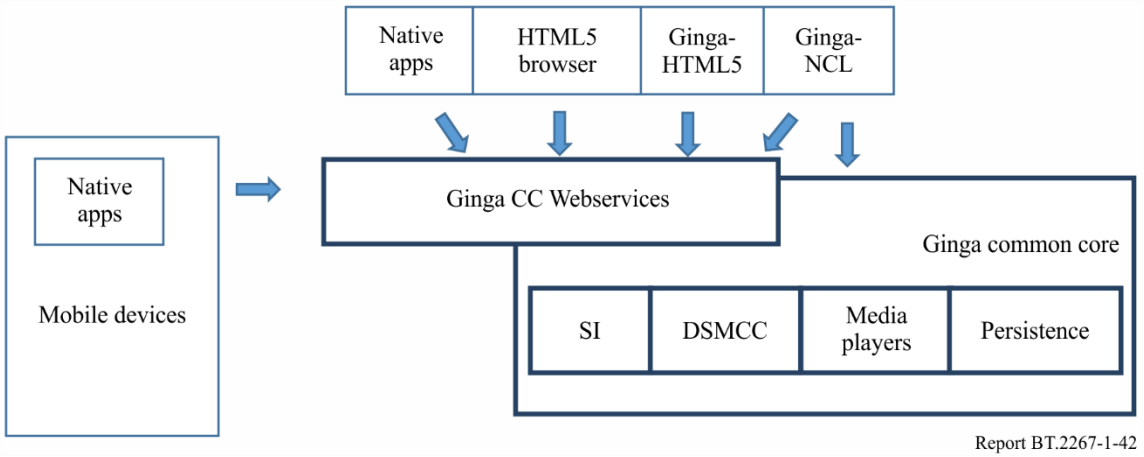
- to access the remote API server, it must be authorized by the user;
- to access the broadcast content, it must be authorized by the broadcaster;

- if it is running on other devices present in the home environment, it must be paired with the interactive DTV receiver.

A simple model of an interactive DTV receiver which includes Ginga CC Webservices is depicted in Fig. 1-50.

FIGURE 1-50

**Interactive DTV Receiver model with Ginga CC WebServices**




## 6 Application control

In Ginga, application signalling, and life-cycle control are delivered by using an application information table (AIT), in accordance with ABNT NBR 15606-3. The AIT can signal applications bounded to the DTV Service or Service Associated Applications (delivered via broadcast or broadband). The signalling and life-cycle control are managed by the selected broadcaster.

Additionally, applications can be launched by Stream Event editing commands, user selection among the available applications (installed or signalled) via Application Catalogue UI, via Ginga CC WebServices app control API or via Ginga CC WebServices deep link API.

The core of an NCL presentation engine is composed of the NCL player and its private base manager module. The NCL player is in charge of receiving an NCL application and controlling its presentation, trying to guarantee that the specified relationships among media objects are respected. The player deals with NCL applications that are collected inside a data structure known as *private base*. NCL applications in a private base may be started, paused, resumed, aborted, stopped and may refer to each other.

*NCL editing commands* control private base operations (openBase, activateBase, deactivateBase, saveBase, and closeBase commands). A second subset of NCL editing commands allow for application manipulation in a private base (to add, remove, and save an application in an open private base; and to start, pause, resume, and stop application presentations in an active private base). A third subset defines commands for live editing in an open private base, allowing the creation of live applications.

NCL editing command to add an application in an open private base also carries information of accessible network domains, including permission information for each domain. Permission information represents allowable access to resources in a network domain.

Applications which are controlled by NCL editing commands coming in the broadcast signal are called *broadcast-oriented managed applications*. These applications are allowed to access broadcast resources of the tuned channel and Web services based on permission information in the *add* NCL editing command.

Ginga also allows for using the application information table (or AIT) and its descriptors, to control broadcast-oriented managed applications, besides the use of NCL editing commands. Each service that has an application associated with it may contain an AIT, and each AIT can contain a description of every application that can run while that service is being broadcasted, and must carry information of accessible network domains with permission information of each domain. For each application_control_code of an AIT there are equivalent NCL editing commands to perform the same tasks.

Applications that are not controlled by NCL editing commands coming in the broadcast signal may still be authenticated by broadcasters by means other than broadcast signals. Permission to access to broadcast resources is given by application authentication.

Ginga associates at least one private base, the *default private base*, with each IBB service. An IBB service may be a broadcast TV channel, or an IPTV channel, or a Web service. When an IBB Service is open, e.g. when a broadcast TV channel is tuned, its corresponding default private base is opened and activated by the Private Base Manager. Other private bases can then be opened (or created), but at most one base associated with each service of an IBB service (for example, a service of a broadcast TV Channel). When an IBB service has just one service, it only has one private base associated, the default private base.

NCL resident applications are managed in a specific private base.

*NCL Editing commands* are wrapped in a structure called event descriptors. They may be issued externally to an NCL application execution, or internally by the execution of an NCLua object.

NCL Editing Commands sent from an IBB service can control only the private bases created by previous NCL editing commands sent through the open IBB Service and the default private base associated with the IBB Service. In other words, editing commands coming from an IBB Service have no effect on private bases associated with other IBB Services. For example, editing commands coming from a tuned broadcast TV channel have no effect on private bases associated to other TV channels.

NCL editing command events generated by NCLua objects running in private bases may control just these private bases.

## 7 Developing and testing Ginga and applications

Ginga has an open-source reference implementation that can be downloaded from http://git.ginga.org.br. Several commercial releases are derived from this reference implementation.

At http://testsuite.gingancl.org.br, Ginga developers and testers can find a comprehensive specification of testing procedures for the Ginga-NCL presentation environment. This helps manufacturers verify the compliance of their products and give confidence that a proper interoperability level can be achieved.

For testing the more TV-related NCL applications, open source NCL Players are available for Windows, Linux and MAC platforms (http://www.gingancl.org.br/en/ferramentas). There are also NCL Players running natively or as plug-ins of HTML browsers.

There are several tools to help authors to design NCL applications. For example, one can find NCL Composer and NCL Eclipse at http://www.ncl.org.br/en/autoria, which are two open-source authoring tools available in the official site of the NCL language.

## References

ITU-T H.761 Recommendation ITU-T H.761, Nested Context Language (NCL) and Ginga-NCL for IPTV Services.

ABNT NBR 15606 ABNT NBR 15606, Digital Terrestrial TV – Data Coding and transmission specification for digital broadcasting.

# Annex 5

# TV Open Platform for Smart media (TOPSmedia) system

## 1 Introduction

TOPSmedia [1] is open smart TV platform standard to define the web runtime environments for smart TV application. The specification includes the state-of-the-art HTML5 technologies as web core and it also defines the extended APIs to control smart TV specific features like channel, programme, device and so on. An application in compliance with this standard is developed and released taking advantage of HTML5 features and interfaces and will provide the same user experience on various receivers; terrestrial, cable, satellite and IPTV.

Figure 1-51 shows the service overview of smart TV applications complied with this specification. The linear AV contents are delivered to smart TV receiver through the broadcast channel. On the other hand, the application and non-linear AV contents are transmitted through broadband channel. A smart TV application can be launched by application control signal from broadcasting provider. And it can be also launched by user after downloaded and installed from app store. The general application developer as well as broadcasting provider can develop smart TV application and register the application to specific app repository.

FIGURE 1-51

**The service overview of smart TV application**

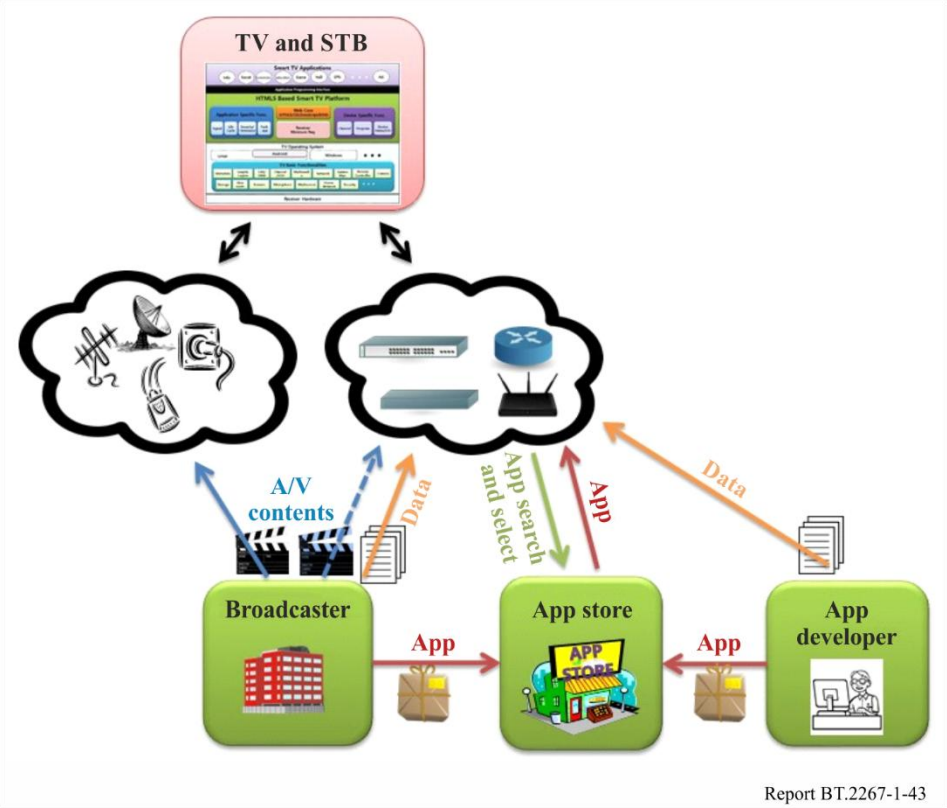
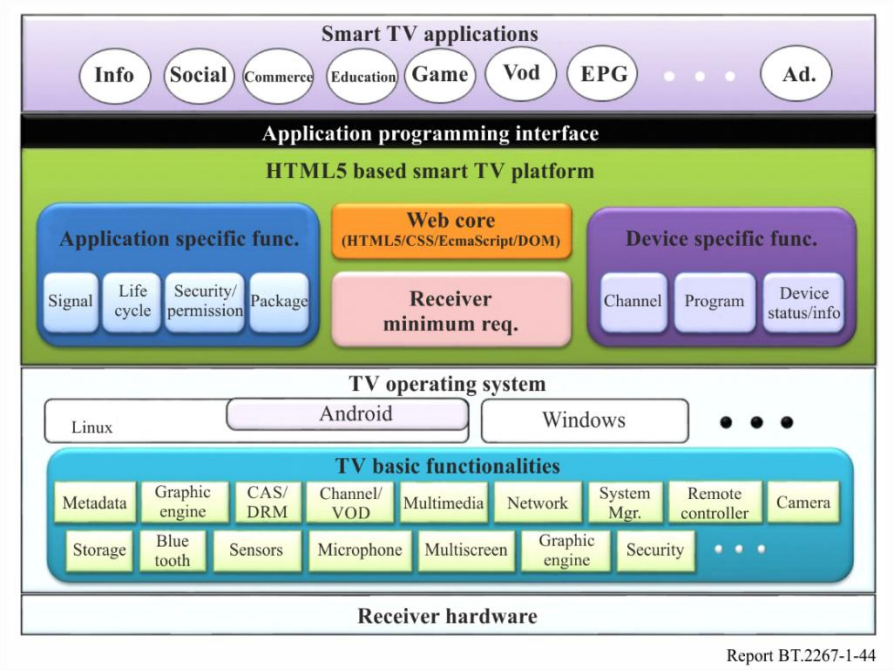



Figure 1-52 shows the architecture of smart TV receiver based on this specification. Smart TV platform supports web runtime environments on TV's basic hardware and software systems like tuner, channel manager, decoder, demultiplexer and so on regardless of TV operating system. With web browser core functionalities, the platform supports various extended APIs to control smart TV device and to manage smart TV application. On the smart TV platform, various applications such as VoD, game, education, information and social can be executable utilizing HTML5 web technology and extra features supported by this specification.

FIGURE 1-52

**The architecture of smart TV receiver**




## 2 Application Model and Lifecycle

In the TOPSmedia, various application's types are defined according to the following criteria:

- Execution Method

Smart TV application can be divided according to execution method.

| Type | Description |
|---|---|
| Store App | This application is launched by user after it was downloaded and installed on the smart TV device through app repository. |
| Signal App | This application is launched by broadcasting signal of Application Information Table (AIT) provided by broadcasting service provider. |
| Broadband App | This application is launched by API call defined in smart TV Extended APIs; createApplication. |

- Application Packaging

Smart TV application can be divided according to whether or not the application is packaged to specific format.

| Type | Description |
|---|---|
| Package App | All application resources are compressed to specific packaging format. This application is downloaded and installed through app repository. |
| Non-package App | This application is not packaged. It can be accessed by application URL as it is usually hosted by web server. |

- Relation with Broadcasting Service

Smart TV has basic functionality as TV receiver. It shall receive linear AV contents from broadcaster and present the contents on the TV screen. When smart TV application is launched, the TV functionality shall be decided whether or not it is still alive like below.

| Type | Description |
|---|---|
| Broadcast Activated App | When this application runs, the broadcasting resources (e.g. decoder, demultiplexer, SI filter) keep still their status. Application shall be running with broadcast service together. |
| Broadcast Inactivated App | When this application runs, the broadcasting resources shall be suspended. Application shall be running alone without broadcast service. |

- Channel Bound

When the application is running as the broadcast activated app, TV channel can be changed by application's channel API call or user's button press of remote controller. At that time, the application can be divided according to whether or not the application keeps the status on the new channel.

| Type | Description |
|---|---|
| Channel Bound App | When TV channel is changed, this application is ended automatically. |
| Channel Unbound App | When TV channel is changed, this application is still running in the new channel. |

The definition for application types above are summarized like below considering the application relation for each other for definition criteria.

<table>
<tr><th>Execution</th><th>Package</th><th>Broadcast</th><th>Channel</th></tr>
<tr><td>Store App</td><td>Package App</td><td>According to “apptype” attribute</td><td>Channel Unbound App</td></tr>
<tr><td>Signal App</td><td rowspan="2">Non-package App</td><td>Broadcast Activated App</td><td>According to “service_bound_flag”</td></tr>
<tr><td>Broadband App</td><td>Broadcast Inactivated App</td><td>N/A</td></tr>
</table>

Meanwhile, according to application's lifecycle policy in this specification, smart TV application can be started like below cases:

- User selection for store app that is already installed on the smart TV device.
- API call by the current running application (createApplication).
- Application signal (AUTOSTART) in broadcasting stream made by broadcasting service provider.

On the other hand, smart TV application can be ended like below cases:

- Channel change while channel-bound application is running.
- API call by current running application (destroyApplication).

- Application signal (KILL) in broadcasting stream made by broadcasting service provider.
- Application priority (low-priority application shall be ended when higher application is started).
- EXIT key on remote controller is pressed by User.
- Smart TV exceptions or unexpected cases (ex. system memory full).

## 3 W3C Profile for Smart TV

TOPSmedia supports application execution based on W3C web environments. However, considering smart TV device's features like screen resolution and input method, some W3C specifications are not suitable to TV environment. So, a W3C profile needs to be made for TV and this specification defines W3C profile based on [OIPF TV PROFILE] [2].

[OIPF TV PROFILE] shows a profile definition of HTML, CSS and other related web technologies aimed to TV services and devices. Its goal is to describe a common profile that can be relied on by content and service providers and implemented by manufacturers.

Therefore, this specification accepts [OIPF TV PROFILE] as W3C web core standard including a few of extra profiles like the following:

- Virtual KeyCode Definition for remote controller input.
- Geolocation API supported.
- Media APIs profile to handle broadcasting video.

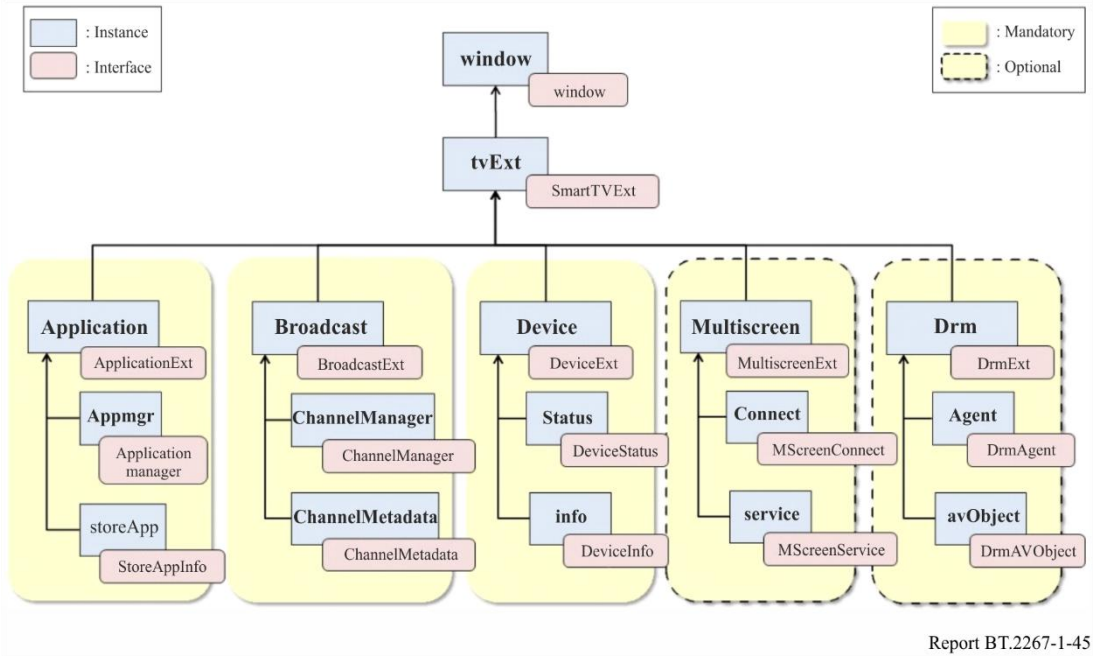
## 4 Extended APIs

TOPSmedia defines the Extended APIs. The Extended APIs are the set of interfaces to support smart TV specific features such as smart TV application, broadcasting resources, smart TV device and other advanced functionalities. Even if these kind of APIs are needed surely to develop smart TV application, they are not defined in W3C specification.

The Extended APIs are defined like in Fig. 1-53. They have the hierarchy structure from Window instance as a root. All interfaces are described according to [WebIDL] [3].

FIGURE 1-53

**Smart TV Extended API structure**




First of all, application interface is defined to support the management for the current running application; application create/destroy, application show/hide, control for keys and permission, and so on. And it also supports the APIs to get the detail information about store apps that are installed on the current device such as id, name, version and description.

Next, broadcast interface is defined to get information about smart TV specific features; channel and programme. The smart TV application can get and utilize the various attributes of channel such as name, number and sourceURI, and the attributes of programme such as name, description, start time, duration, an element audio/video and so on. In this specification, the channel change operation can be done using HTML5 <video> element; sourceURI value is assigned to 'src' attribute of <video> element.

And device interface supports getting the detailed information for the device which the application is running on. The information includes the device's capability and network configuration as well as device specific attributes like manufacturer, model name, HW or SW version and so on.

Finally, multiscreen and DRM interface are defined to support smart TV advanced features. Utilizing multiscreen interface, the application can be connected to the companion devices like smart phone and tablet and then it can communicate with the target application running on the specific companion device. DRM interface provides the method to communicate with DRM agent of the device and to handle DRM-related information of object playing DRM-protected contents. In this specification, DRM interface is based on [CEA-2014B] [4] specification.

## 5 Application Signalling and Packaging

TOPSmedia supports the application's lifecycle control according to application signal provided by broadcaster. Application signal information exists in AIT (Application Information Table) section in the broadcasting stream. In this specification, application signalling definition is based on [DVB SIGNAL] [5]. The definition of "application_signalling_descriptor" that is used in PMT (Program Map Table) to indicate AIT section is complied with clause 5.3.5.1 of [DVB SIGNAL].

And the followings among AIT descriptors defined in [DVB SIGNAL] are supported in this specification:

- application descriptor;
- application usage descriptor;
- transport protocol descriptor;
- simple application location descriptor;
- application name descriptor.

As application delivery protocol, this specification support HTTP only and it does not support application delivery through DSMCC.

On the other hand, this specification also supports the application's lifecycle control by user in case of store app. All store apps (package apps) distributed and installed through app repository shall be compressed and packaged by specific format. The package requirements of this specification are based on [W3C Widget] [6]. Additionally, this specification defines some features and new elements considering smart TV environments.

This specification defines new URL scheme to use it as "id" attribute's value of widget element.

**tvapp://widget.organization_id.application_id**

Store app (package app) shall have id as the pair of two values to identify it; organization_id and application_id. The two values shall be defined uniquely according to the specification's rules.

And the new elements are defined like below:

- <apptype> element
  - to define application type related to application execution method
  - to indicate whether or not it can be launched without network connection
- <appdomain> element
  - to support the application can access any resources in specific domain under the same-origin policy.

## 6 Other features

TOPSmedia defines protocols and contents formats exactly that shall be supported by smart TV receiver. As a protocol for application delivery and download, it supports HTTP 1.0/1.1 and HTTPS. And it supports HTTP 1.1, RTSP, MPEG-DASH profile for streaming contents. Also, graphic formats, broadcast contents format, broadband contents formats and so on are defined in detail.

Meanwhile, the specification defines the receiver minimum requirements that all smart TV receiver shall be complied with. The requirements consist of screen resolution, colour format, text input method, video scaling factor, font, input devices and so on.

Finally, the specification includes the own profile definition considering the device's target services based on device capabilities, broadcasting environments, business model and so on; full profile, broadcast related profile, basic package profile and advanced package profile. They include the implementation coverage and the specific restrictions for this specification.

## 7 Service examples

In this section, a few of smart TV applications complied with TOPSmedia are introduced as a service example. The following screenshot shows the broadcast activated application that displays the broadcasting related information with broadcast video. When user selects the current channel, the application is running automatically with red button image according to broadcaster's signal. If user presses the red button on remote controller, the application is changed like the screenshots.

FIGURE 1-54

**Service example of the broadcast activated and signal app (broadcasting info)**




Report BT.2267-1-46

The next is the application for audio channel. Originally, in the audio channel, there is no broadcasting video and there is just black screen instead. When user selects the channel, the application is running with a variety of the audio information such as title, singer, lyrics, next song, current playing time and so on.

FIGURE 1-55

**Service example of the broadcast activated and signal app (audio channel)**




Report BT.2267-1-47

The following screenshot shows a ticker application. User can select and launch the application through receiver's app store UI. When the application is running, the broadcasting video is still alive because it is configured as the broadcast activated app. The application shows the real time news as a text string in the bottom and it plays the VoD stream of the news if user presses blue key.

FIGURE 1-56

**Service example of the broadcast activated and store app (ticker app)**

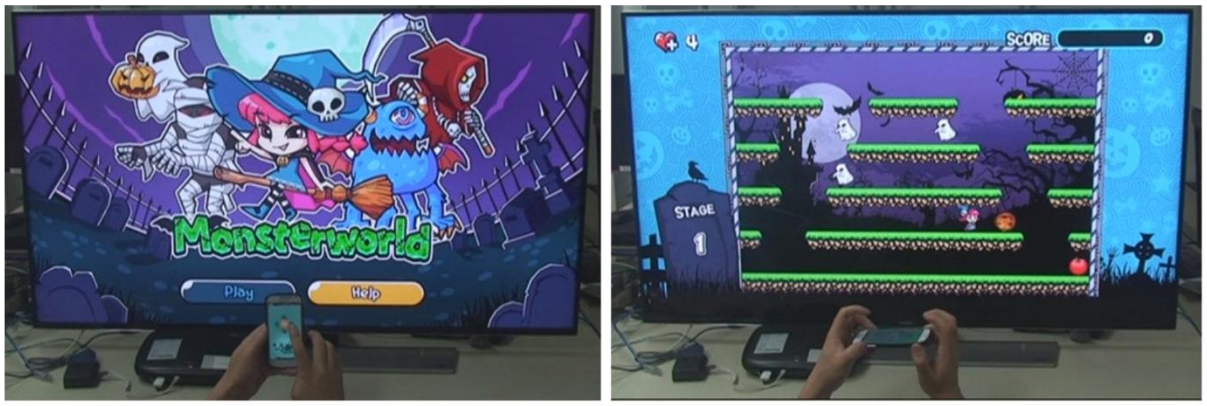



Report BT.2267-1-48

Finally, the following screenshot shows the multiscreen application for smart TV game. User can start the smart TV game application through receiver's app store UI and the user can launch the application of the mobile device together. Smart TV application discoveries the mobile device and connects with it using the multiscreen extended APIs. User can control the game's character through controlling smart phone. A variety of interfaces such as orientation, swipe, touch and shake are delivered to smart TV application to control the game.

FIGURE 1-57

**Service example of the broadcast inactivated app and store app (multiscreen game)**




Report BT.2267-1-49

## 8 Future works

TOPSmedia has been developing continuously at the related TTA standardization committee for the next revision. The next version will include new features like contents synchronization, remote application control, advanced user input like gesture and voice, T-commerce and so on. It is expected to be released at the next year.

**Attachment**: Cloud TV Open Platform for Smart media (COPSmedia) system

# Attachment
# to Annex 5

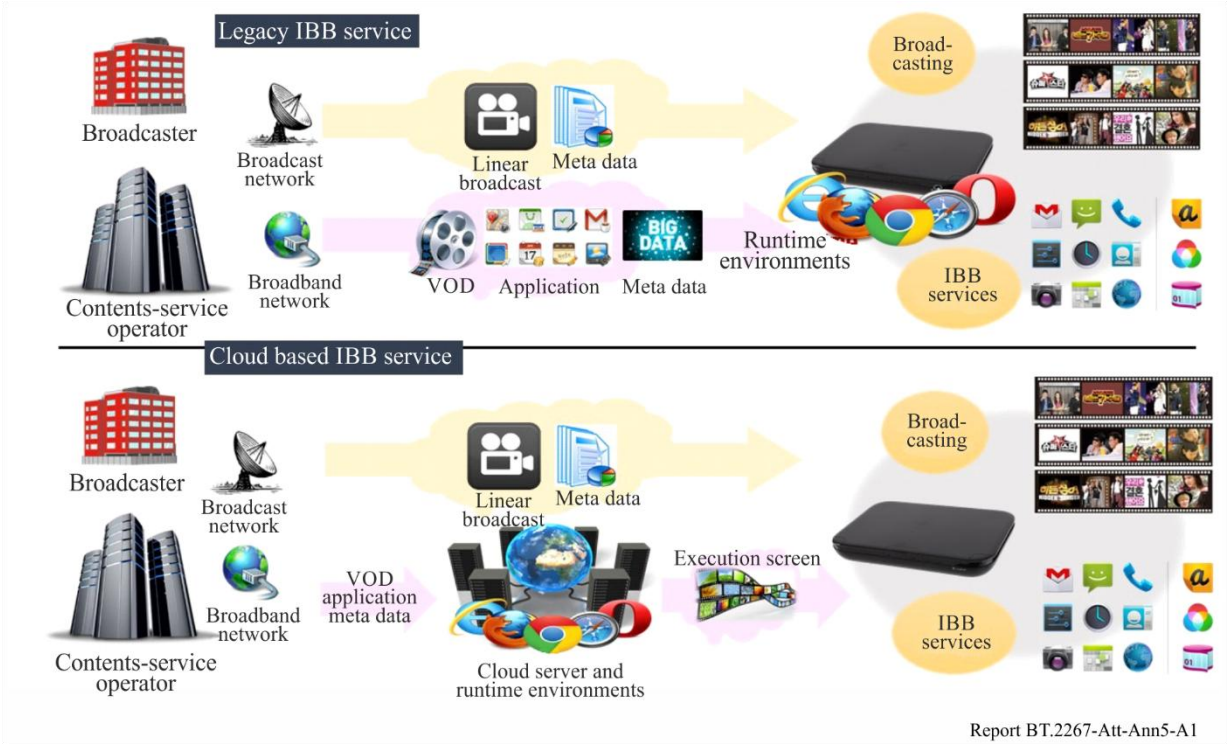
# Cloud TV Open Platform for Smart media (COPSmedia) system

## 1 Introduction

Cloud broadcasting is a kind of virtualization service based on cloud streaming technology for the Integrated Broadcast-Broadband (IBB) service. Instead of the broadcasting receiver, the cloud server runs IBB services using powerful computing resources like CPU, GPU and memory and then it sends the display screen and information to the receiver. Receiver just needs to present the display screen without any application runtime environments for the IBB services based on smart TV platform. It is very useful solution considering the low-performance receivers like an outdated (old-model) set-top box. Actually, major Cable and IPTV broadcasters in Korea have had the difficulties of the smart service's expansion for their customers because many customers still use the old set-top boxes they cannot afford to have an application runtime for IBB services. Figure 1-58 shows the comparison between a legacy and cloud broadcasting for the IBB services.

FIGURE 1-58

**The comparison between legacy and cloud IBB service**




In Fig. 1-58, the smart TV platform can be realized on cloud server-side and it is not needed to be implemented into broadcasting receiver anymore. So, the outdated receivers can be smart without any hardware and software upgrades because they do not need to have any application runtime environments for IBB services.

TTA as standard organization in Korea has developed the cloud broadcasting specification called Cloud TV Open Platform for Smart media (COPSmedia). The official title of the standard is "Open Cloud-based Smart Broadcasting Platform, TTAK.OK-07.0129" [7]. It has almost the same IBB runtime environments with TOPSmedia, so all IBB services of smart TV platform can be running on cloud broadcasting platform with interoperability between two standards.

FIGURE 1-59

**The architecture of cloud broadcasting service**

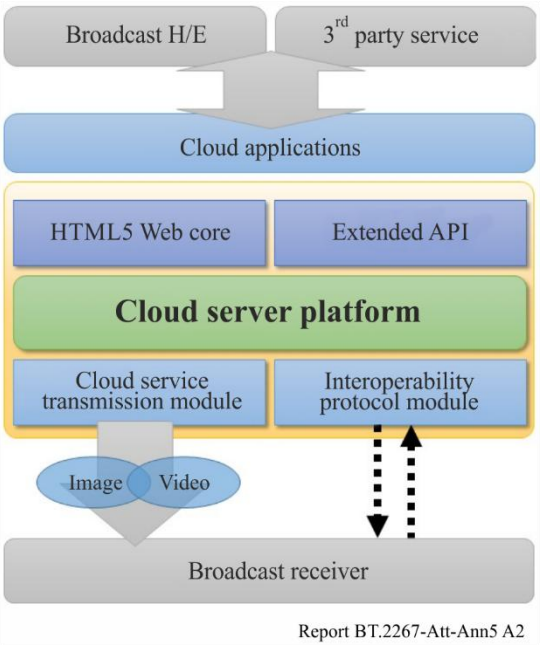



Figure 1-59 shows the architecture of cloud broadcasting service according to this standard. Basically, cloud server platform supports HTML5 web core and an extended interfaces for cloud IBB application's runtime environments. And it has a cloud service transmission module to send the display screen to receiver and an interoperability protocol module to handle some interactions by API call between the server and receiver.

## 2 TOPSmedia profile

COPSmedia includes a lot of features from TOPSmedia as the application runtime environments. Cloud server platform compiles with HTML5 web core, application model (type and lifecycle), extended API, receiver key control and so on. So, all IBB applications can be executed on cloud server platform without any changes because they keep compatibility between them. It means the runtime environments for IBB services move from a receiver-side to a cloud server-side. In addition, cloud server platform can support any new feature optionally to utilize the high-performance resources like 3D, storage, metadata and so on.

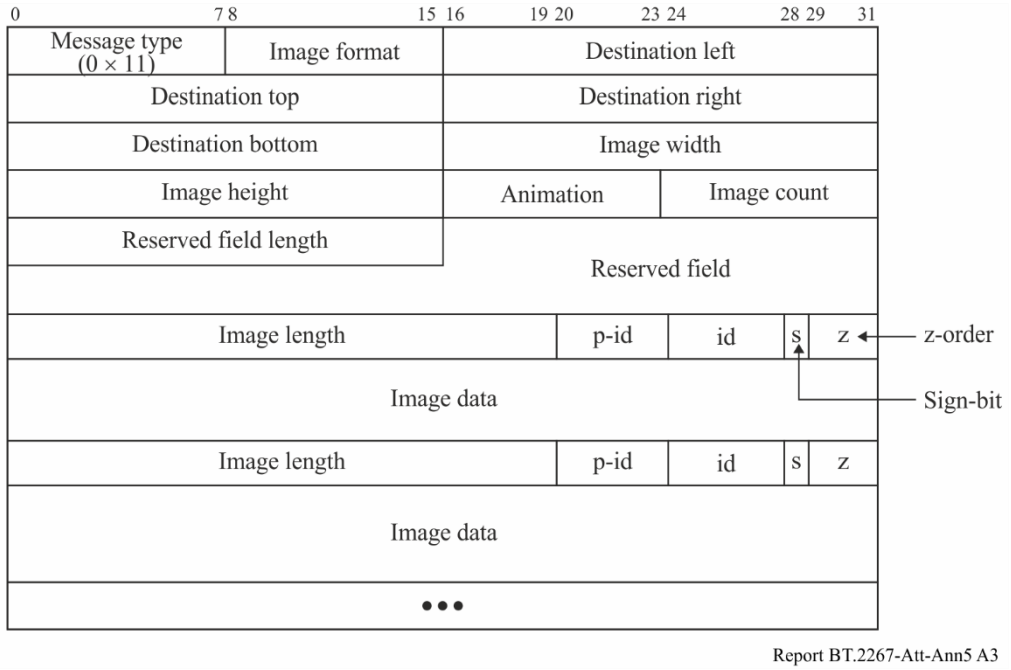
## 3 Cloud service transmission

To present a cloud IBB services on the receiver, the data transmission for the services is needed from cloud server to receiver. In COPSmedia, there are two major methods for it; image-based transmission and video-based transmission. Image-based transmission means the cloud server sends the captured image data from the display screen on the server. It is usually partially updated data for the changed area at any one moment. On the other hand, video-based transmission means the entire display screen is captured and encoded to video format and then it is sent to the receiver through streaming protocol.

Cloud broadcasting service provider can select the transmission method according to their cloud service applications. For example, the one is suitable and efficient for the applications having the partial change on the screen like UI menu navigation and the other is suitable for the application having the full change like VOD and game. COPSmedia defines the data format for cloud service transmission in application protocol layer. (It is over the common transmission protocol layer like TCP or UDP). Figure 1-60 shows the example of data format for image-based transmission.

FIGURE 1-60

**The example of data format for image-based transmission**

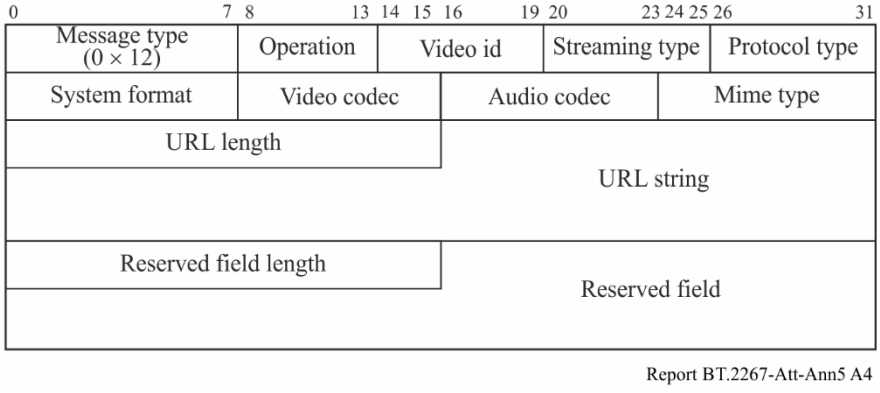



In Fig. 1-60, the image data format consists of the metadata for the updated image area and the actual image data. In the metadata fields, there are the format, position, size and effect information to update partial area on the receiver screen.

Meanwhile, Fig. 1-61 shows the example of data format for video-based transmission.

FIGURE 1-61

**The example of data format for video based transmission**



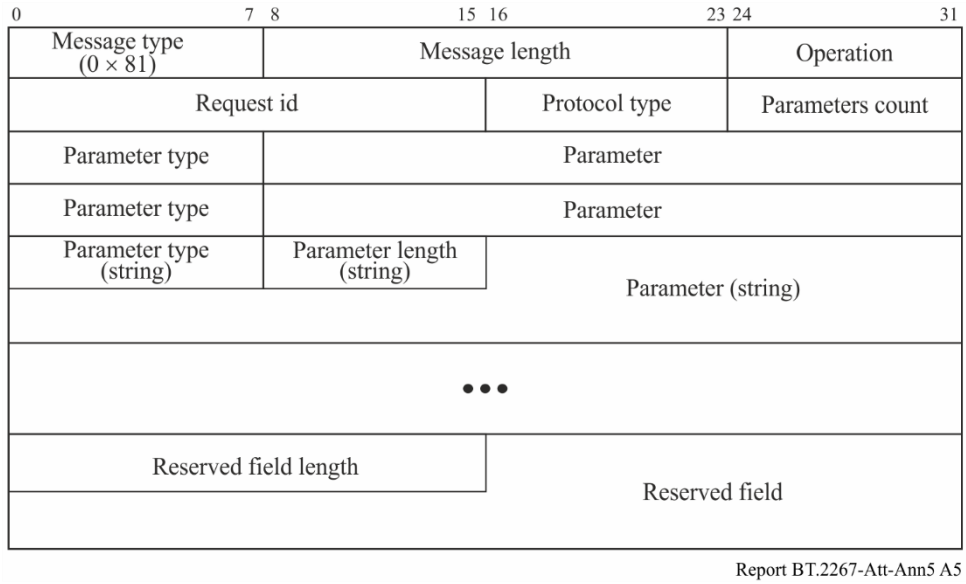
In Fig. 1-61, the video data format has the video URL and streaming information like operation, type, format and codec. Because this format does not include the actual video data itself unlike image-based transmission, the receiver can get the video data from VOD server targeted by URL.

## 4 Interoperability protocol

In COPSmedia, an IBB application for smart service shall be launched and run on the cloud server-side. It is not on the receiver-side. Sometimes, the cloud application needs to get and control any receiver specific information like channel, program, customer (device) data and so on. In this case, the information shall be delivered from the receiver to the cloud server according to application's request. To support this mechanism, this standard defines the interoperability protocols like below. Figure 1-62 shows the data format definition for interoperability protocol.

FIGURE 1-62

**The data format for interoperability protocol**




The interoperability protocol is related to an extended API's function call and return values. In Fig. 1-62, the parameter type and parameter fields correspond to the actual function parameters for the specific extended API. For example, to support the cloud IBB application to get customer code from receiver, the field values of data format are decided like in Fig. 1-63.

FIGURE 1-63

**The example of field value definition in the interoperability protocol**

■ **REQUEST**

| Field | Value | Meaning |
|---|---|---|
| Message type | 0 × 81 | Interoperability protocol |
| Message length | (variable) | Full length (BYTE) |
| Operation | 0 × 1 | REQUEST |
| Request id | (variable) | Unique id for this request |
| Protocol type | 0 × 71 | Customer code request protocol |
| Parameters count | 0 | The number of parameter |

■ **REPONSE**

| Field | Value | Meaning |
|---|---|---|
| Message type | 0 × 81 | Interoperability protocol |
| Message length | (variable) | Full length (BYTE) |
| Operation | 0 × 2 | RESPONSE |
| Request id | (variable) | Corresponding request ID |
| Protocol type | 0 × 71 | Customer code request protocol |
| Parameters count | 1 | The number of parameter |
| Parameter type | 0 × 04 | String type |
| Parameter length | (variable) | String length |
| Parameter | (variable) | Customer code value |

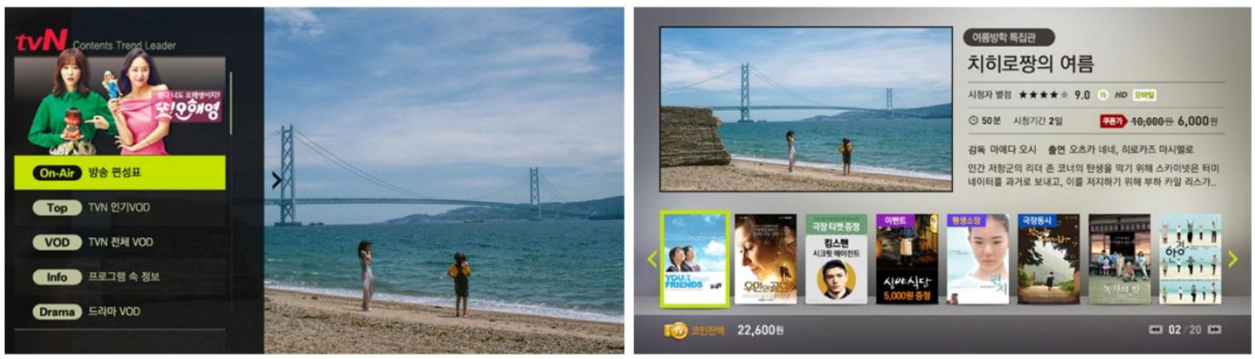
Report BT.2267-Att-Ann5 A6

## 5 Service example

Major Cable and IPTV broadcasting operators in Korea have served or prepared the various IBB application services based on COPSmedia. Actually, over 4 million subscribers are using the various cloud services without the replacement of outdated set-top box receiver. Figure 1-64 shows the first example of cloud IBB service complied with COPSmedia. The type of this application is the broadcast activated application that is running with the linear broadcast service. In this application, the image-based transmission is used for TV main UI and VOD UI because the application is updated partially and it shall be executed with linear broadcast.

FIGURE 1-64

**The first example of cloud broadcasting service (broadcast activated application)**




Report BT.2267-Att-Ann5 7

On the other hand, Fig. 1-65 presents the second example of cloud IBB service. The type of these applications is the broadcast inactivated application that linear broadcast service is suspended at the moment the application is started and the applications are running independently without the linear broadcast service. The example services in the Figure are VOD and game. Because this kind of applications are updated on the entire area of screen, the video-based transmission is used for them.

FIGURE 1-65

**The second example of cloud broadcasting service (broadcast inactivated application)**




Report BT.2267-Att-Ann5 A8

If these applications are running on the broadcasting receiver side, they require high-performance resources like CPU, GPU and memory, especially in case of IBB game service. However, they are running on the cloud server side and the receiver just receives and presents the display screen based on this standard. So, they can be running fast on the very low-performance receiver like under 1000 DMIPS.

## References

[1] TTAK.KO-07.0111/R1, “HTML5 based Smart TV Platform”, April 2014.

[2] Open IPTV Forum Feature Package Specification, “Volume 5a – Web Standards TV Profile [V1.0.0]”, April 2013.

[3] W3C, “Web IDL”, April 2012. Available from http://www.w3.org/TR/2012/WD-WebIDL-20120207/

[4] ANSI/CEA-2014-B, “Web-based Protocol and Framework for Remote User Interface on UPnP™ Networks and the Internet (Web4CE)”, January 2011.

[5] ETSI TS 102 809 V 1.1.1, “Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in hybrid broadcast / broadband environments", January 2010.

[6] W3C, “Widget Packaging and XML Configuration”, September 2011. Available from http://www.w3.org/TR/2011/REC-widgets-20110927/

[7] TTAK.KO-07.0129, “Open Cloud-based Smart Broadcasting Platform”, December 2016.

PART 2

# General considerations to build IBB services

TABLE OF CONTENTS

*Page*

1 Consideration on use of integrated broadcast-broadband technologies on mobile devices ..... 86

1.1 User requirements ..... 86

1.2 Interoperability between telecommunication services and interactive broadcasting services ..... 87

2 On-demand content playback ..... 87

2.1 Consideration on behaviour of on-demand content playback ..... 87

2.2 Use of MPEG-DASH ..... 88

## 1 Consideration on use of integrated broadcast-broadband technologies on mobile devices

These user requirements apply when the mobile consumer device is designed to be able to simultaneously connect to broadcast emissions and to the Internet by means of separate over the air connections.

### 1.1 User requirements

When considering the user requirements for mobile reception of broadcasting of multimedia and data application, the following should be considered:

– access to content and services may be controlled via conditional access/service access protocols and other content protection mechanisms;

– seamless service access to content and services across networks;

– flexible configuration of a large variety of services;

– transmission of good quality of image, video, audio and data services;

– support for interactivity, where content and applications have to be interactive;

– technical aspects enabling interoperability of the services between broadcast and telecommunication networks, for example, content format, audio/video codecs, and encapsulation methods.

TABLE 2-1

**Technical characteristics of mobile reception include**

| User requirements | Technical characteristics |
|---|---|
| a) Image resolution, frame rate and bit rate | i) Video should have QVGA, WQVGA as well as other display resolutions. Up to ~2.25 Mbit/s per stream as frame rate and up to 30 fps as bit rate.<br>ii) Audio should have stereo and mono, 12 kbit/s bit rate supported.<br>iii) Data supported include binary data, text, data, audio/video file distribution |
| b) Monomedia coding | a) Video:<br>– H.264/AVC.<br>b) Audio:<br>– HE AAC-v2<br>c) Data format<br>– MPEG4 files.<br>– JPEG. |
| c) Flexible configuration of services:<br>– Audio/video<br>– Ancillary and auxiliary data | – Real-time audio and video.<br>– Scheduled content and file download based on network load.<br>– IP data streams.<br>– Electronic programme guide.<br>– $C/N$ = 13.5 dB in a typical urban mobile environment. |

### 1.2 Interoperability between telecommunication services and interactive broadcasting services

In order to achieve the interoperability of telecommunication services and interactive broadcasting services, two levels have to be considered; one is content level and second is service level, which have to complement each other in order to have interoperability of those two media of transmission of video and audio signals.

The content format level of multimedia format of ISO/IEC 14496-14, mono-media coding of audio coding of ISO/IEC 14496-3/2001:Amd.4, video coding of ISO/IEC 14496-2/10 MPEG-4 AVC and data coding of ISO/IEC 10918 (JPEG) are possible formats.

## 2 On-demand content playback

### 2.1 Consideration on behaviour of on-demand content playback

When playing back on-demand content in IBB systems, an IBB application can control to playback on-demand content. Although control of playback of on-demand content by an application is similar to playback of on-demand content on web site, these two playbacks are not the same. In contrast to playback of on-demand content on the web, there are two technical elements to consider to playback on-demand content under the control of the IBB application; status of a receiver to receive broadcast signal, and lifecycle of the application. To handle these technical elements, two scenarios can be considered to play back on-demand content in IBB systems. They are simple playback and playback with an associated IBB application. In some IBB systems, such as Hybridcast, these two scenarios are supported.

Simple playback is the scenario in which the same IBB application runs during playback of the on-demand content. For example, an IBB application starts playing back of on-demand content and controls its presentation during playback. When stopping playback of on-demand content, the IBB application controls how to present the stopping content and things coming next, including showing live broadcast programme. This scenario is shown in (a) of Fig. 2-1. If the IBB application is designed to offer services specific to the playing on-demand content, viewers will be able to enjoy the IBB services. However, for such IBB services, it is required that the IBB application is implemented to provide such functionalities beforehand. In addition, service providers who implement the IBB application in this way may require careful operation to offer the IBB services for on-demand content because the service provider should take care about two services, the IBB service which starts that IBB application and the service closely related to the on-demand content, at the same time.

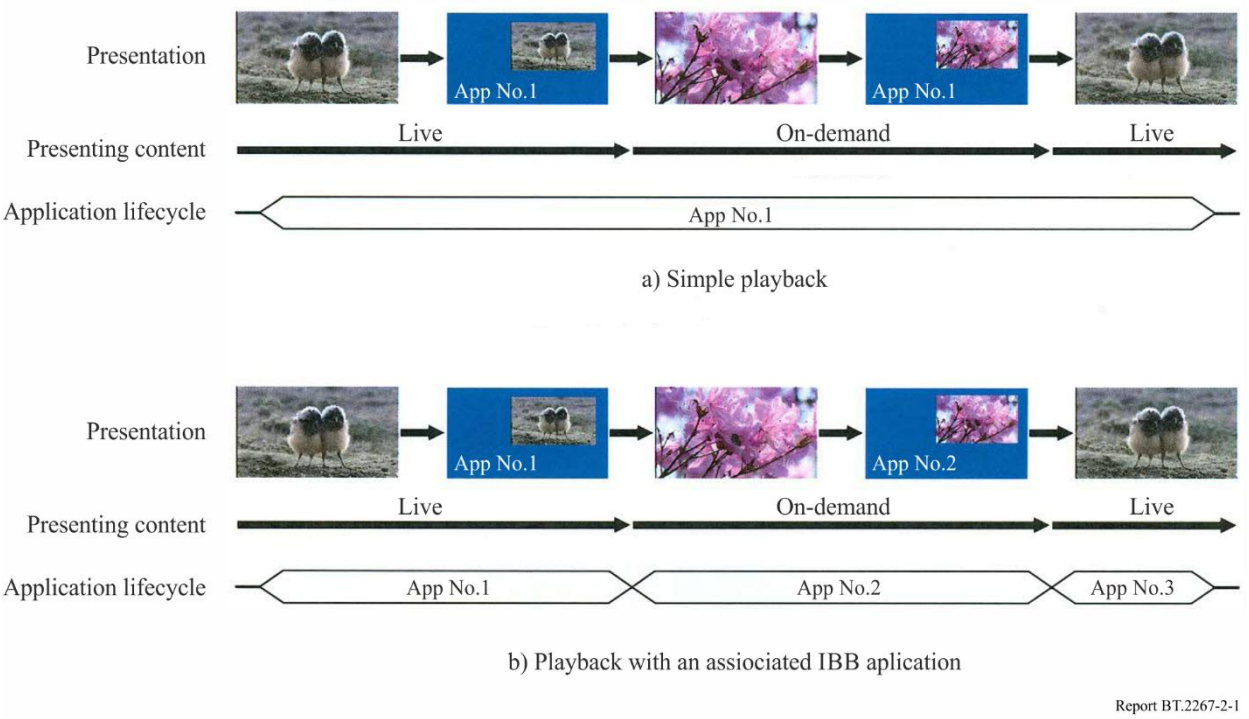
For playback with an associated IBB application, the IBB application is altered to a programme associated IBB application. In this case, on-demand content can be presented as if it is a live broadcast programme. If the on-demand content is content for catch-up service, service provider can offer the same IBB service without modifying the associated IBB application. In other words, the associated IBB application can be reused to offer an overall user experience specific to the on-demand content in addition to audio-visual content. This scenario is shown in (b) of Fig. 2-1. In the Figure, App #2 is the application associated to the on-demand content. The App #2 is launched by the application control information in conjunction with start of on-demand content. If the IBB application starting playback of on-demand content is a service associated application[8], this can be achieved by ‘tune’ to an on-demand content. Since the application is launched by the application control information associated with on-demand content, it is natural that the application is terminated at the time of stop

[8] Defined in Recommendation ITU-R BT.2053 as “An application that is part of the integrated broadcast and broadband (IBB) DTV service tuned to by the user at a given time”.

of playback. A receiver will follow application control information available at the time of stop of playback. In (b) of Fig. 2-1, the App #3 is instructed to launch by broadcast signal.

FIGURE 2-1

**On-demand content playback scenarios**




When building and offering IBB services which include on-demand content playback, receiver status for receiving broadcast signals should also be considered. In some cases, broadcast signal is not received by IBB receivers during on-demand content playback. For example, a receiver which provides only one transport stream demultiplexer may suspend reception of broadcast signal during playback on-demand content. When signal format of on-demand content is transport stream, the demultiplexer is occupied to decode the signal of the on-demand content, and no demulitplexer is available to decode broadcast signals. Therefore, the receiver is blinded to any control signals coming over broadcast channels. On the other hand, receiver which is capable to receive both broadcast and on-demand signals simultaneously can follow the application control information coming over the broadcast channels during on-demand content playback. If the application control information instructs to terminate the IBB application which controls playback of on-demand content, the playback of the on-demand content will also be terminated. To avoid unintended sudden termination of on-demand content playback for a user, service provider may require transmitting application control information indicating the IBB application is allowed to continue running even after supposed period of execution. Such indication is represented as ‘PRESENT’ application_control_code in HbbTV and Hybridcast.

## 2.2 Use of MPEG-DASH

MPEG-DASH is an adaptive streaming protocol as defined in ISO/IEC 23009-1. MPEG-DASH allows to select appropriate bitrate depending on broadband environment dynamically and works as a packaging format by flexible description in Media Presentation Description (MPD) file. By these features, use of this protocol is becoming popular in many systems including Recommendation ITU-T H.721. Use of MPEG-DASH is also useful for on-demand content playback in IBB systems. In this clause, consideration on use of MPEG-DASH specifically in IBB systems is provided.

### 2.2.1 Profile of MPEG-DASH

An MPD file is a key information file in MPEG-DASH. An MPD file is an XML formatted file to describe meta-data and structure information of the targeted audio-visual content. MPEG-DASH provides hierarchical structure of audio-visual encoded data as Segment, Representation, Period, and Media Presentation. An MPD file contains information on each layer regarding the content, such as position of each Segment, URL of each Segment, audio channel configuration, used encoding scheme, and size of video image. A media player reads MPD file(s) to playback audio-visual content prior to retrieve actual encoded data file(s), follows the information in the MPD to retrieve appropriate audio-visual encoded data and play them. Since ISO/IEC 23009-1 defines MPEG-DASH in very flexible manner, it is required to define operational profile to constrain the sizes or complexity of various parameters in ISO/IEC 23009-1 to make the system practical. For this sake, ISO/IEC 23009-1 defines several profiles for its use such as ISO Base Media File Format (ISOBMFF) On Demand profile and ISOBMFF Live profile.

DVB and IPTV Forum Japan define MPEG-DASH profiles for actual use for on-demand content playback based on ISOBMFF Live Profile. Information on these profiles are available as:

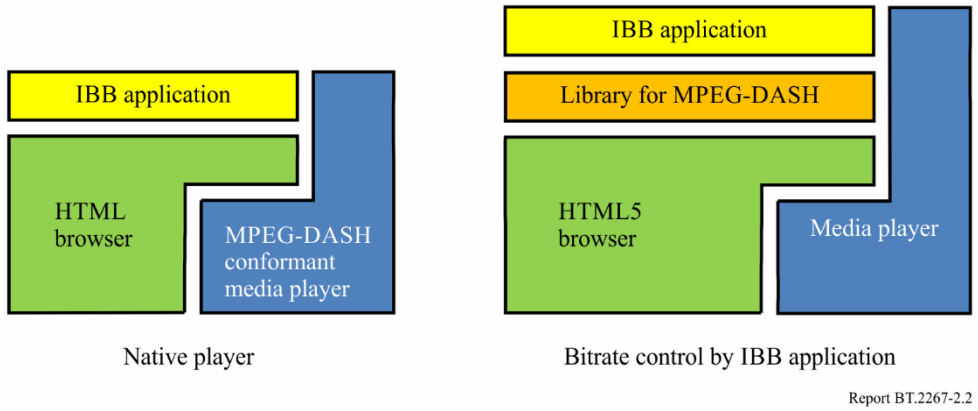
– ETSI TS 103 285 V1.1.1
  http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=44506
– IPTVFJ STD-0013 (in Japanese) http://www.iptvforum.jp/download/

HbbTV uses the profile defined by DVB. Hybridcast uses the profile defined by IPTV Forum Japan. Conformance to these profiles is essential for proper presentation of on-demand content using MPEG-DASH in these systems respectively.

### 2.2.2 Bitrate control in MPEG-DASH

When playing on-demand audio-visual content using MPEG-DASH, it is required to control bitrate of the content. In MPEG-DASH, bitrate control is always under control of a client media player in a receiver. So, control scheme is categorized how a media player in a receiver behaves. There are two ways to control bitrate. Example structures are shown in Fig. 2-2.

FIGURE 2-2
**Example of MPEG-DASH playback structure in IBB systems**




*a) Native player*

In either scenarios in Fig. 2-1, it is natural that an application calls a built-in media player to playback VOD content. Native player approach takes the way to control bitrate within the media player. That is, the application simply designates the URL of VOD content and passes it to the media player. HbbTV with MPEG-DASH takes this approach. This approach has an advantage that the almost the same description in the application can be used for VOD content either by MPEG-DASH or other

protocols. From the viewpoint of an IBB application, both broadcast audio-visual content and on-demand content are just the mono-media to be handled by a media player. This approach makes it easy to 'tune' to on-demand content and thus reuse of an application described in case (b) of Fig. 2-1 can be taken easily. However, in this approach, all the IBB service capable receivers in any kind have to provide the media player which is conformant to the profile used. If the profile includes constraints specific to IBB systems, a build-in media player has to be compatible with such constraints.

*b) Bitrate control by an IBB application*

If the same IBB application runs at the time of start of playback of on-demand content and period of the playback, bitrate control by the IBB application is another approach. For HTML5 application which is intended to control bitrate of MPEG-DASH, W3C is developing an extension called Media Source Extensions (MSE). As of May 2015, MSE is Candidate Recommendation at W3C. MSE consists of a set of APIs with their behavioural models and some extensions of HTML5 elements. Based on MSE, DASH Industry Forum which is a consortium for deployment of MPEG-DASH developed reference client software in Javascript called dash.js, which is intended to work with MSE capable HTML5 browsers on PCs. By development of this client software, it is shown that MSE is feasible. For embedded browsers of which performance is usually lower than that for PCs, the bitrate control library similar to dash.js has been developed to run on Hybridcast receivers. Some test implementations on Hybridcast show consumer IBB receivers provide enough performance to execute this library on MSE capable embedded browsers for HDTV on-demand programmes.

This approach has advantages to deploy MPEG-DASH based services toward cross platform environment. In this approach, a built-in media player which is conformant with MPEG-DASH profile is unnecessary because an IBB application is responsible for bitrate control. That is, a basic media player and MSE capable browser in various devices including IBB receivers, tablet computers and smart phones can be used commonly.

Service providers of IBB services can control presentation of audio-visual content delivered by MPEG-DASH through their own IBB applications. In contrast, because an IBB application to control bitrate has to be started prior to retrieve on-demand content, this approach requires modification of an IBB application or specific IBB applications for case (b) in Fig. 2-1. More information on DASH Industrial Forum, MSE can be found at:

– DASH Industry Forum http://dashif.org/
– Media Source Extensions http://www.w3.org/TR/media-source/

In either ways, use of MPEG-DASH requires careful investigation and consideration on broadband network environment where deployment of on-demand content playback is considered. It is important that choice of available bitrates, encoding schemes, length of period, deployment scenario etc. is well balanced with the environment.

PART 3

# Harmonization of applications among different IBB systems

TABLE OF CONTENTS

*Page*

1 Introduction ..... 92

2 Approaches for harmonization ..... 92

2.1 Possible architecture of harmonized application execution environment ..... 92

2.2 Commonality of application type ..... 94

3 Comparison of application environment of various IBB systems ..... 95

3.1 Comparison of additional objects, properties, and methods ..... 96

3.2 Comparison of application life-cycle ..... 101

4 Harmonization of second screen handling ..... 103

4.1 Communication protocols used between devices ..... 103

4.2 Communication management ..... 104

# 1 Introduction

Recommendations ITU-R BT.1699, ITU-R BT.1722 and ITU-R BT.1889 are Recommendations for harmonizing interactive TV systems and content. Recommendation ITU-R BT.1699 defines the common core of presentation engines, intermediate language for the conversion of interactive content, and the runtime environment required to provide multiple language options. Recommendation ITU-R BT.1722 defines the common core of the execution engines. Recommendation ITU-R BT.1889 defines the high-level architecture of the common core.

As described in Recommendations ITU-R BT.2037 and ITU-R BT.2053, IBB systems combine both broadcast and broadband. By the use of broadband, IBB systems enable more flexible services than interactive TV systems. Further, the use of broadband allows them to easily offer international services.

The IBB systems included in Recommendation ITU-R BT.2075, namely HbbTV(1.5/2.0), Hybridcast/ARIB STD-B62, TOPSmedia and Ginga-HTML5 (ABNT NBR 15606-10), are all HTML-based systems. However, applications in these systems are not compatible, and therefore, it is required to build individual application(s) for the same services in each system. To exchange the IBB services offered by the IBB systems considered in this Recommendation and to deploy these services in an area where other systems are used, it is important to harmonize the IBB systems to ensure compatibility with the IBB applications and interoperability across systems.

Part 3 defines methods to harmonize the IBB systems based on an analysis of these systems from several perspectives.

# 2 Approaches for harmonization

## 2.1 Possible architecture of harmonized application execution environment

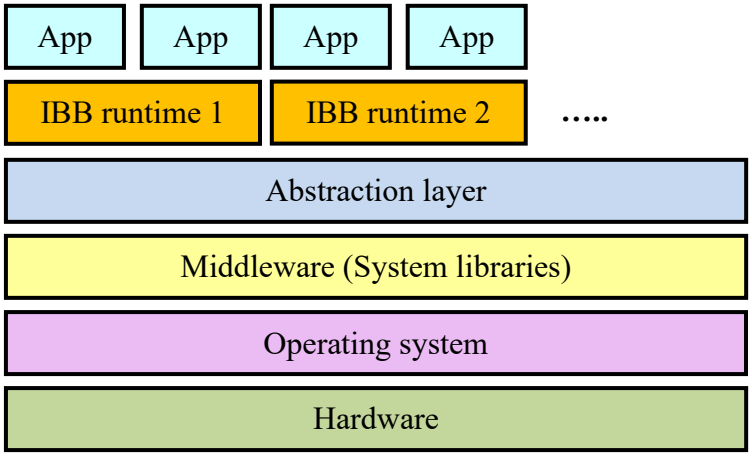
There are several methods to provide an IBB service by using the same application on multiple platforms.

### a) Abstraction of execution environment

This approach is to provide an abstraction layer in the execution environment in a receiver. The abstraction layer unifies the underlying functionalities into a set of APIs so that its upper layer software can be individually implemented on the basis of the implementation of the APIs. Further, the multiple upper layer software can be used for its specific purposes. Figure 3-1 shows how the abstraction layer works for the harmonization of IBB systems.

FIGURE 3-1

**Basic IBB receiver structure with an abstraction layer**




In Fig. 3-1, multiple IBB execution environments (runtime) can be implemented on the same abstraction layer. Compared with that in the case of Recommendation ITU-R BT.2075, each IBB runtime in Fig. 3-1 corresponds to each IBB system included in the Recommendation. Since the IBB systems in the Recommendation are based on HTML, an HTML browser can be considered to play some role of the abstraction layer. In this case, each IBB runtime will be implemented in the form of a JavaScript library. A combination of a web browser and a JavaScript library leads to very powerful expandability as seen in the implementations of the services using Ajax (which stands for Asynchronous JavaScript + XML). However, the functionality of this combination is limited by the functionality of the web browser. In the receiver of IBB systems, IBB specific functionalities which are not normally installed in a web browser are achieved by the middleware, or by a combination of the middleware and a dedicated web browser. Therefore, an abstraction layer, or a web browser, should be designed to allow communication between the IBB runtime and the middleware.

**b) Application conversion**

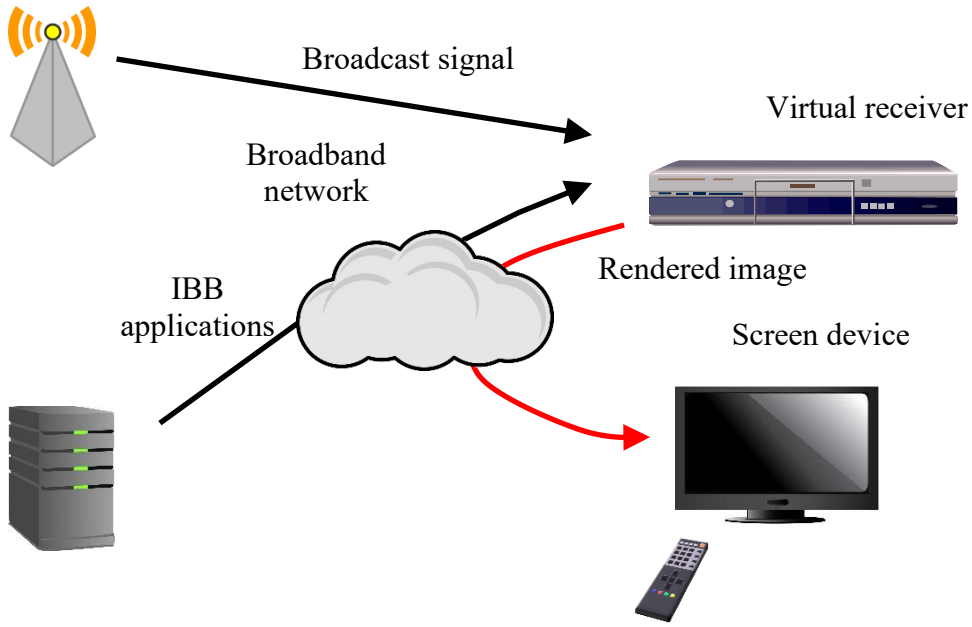
This approach is intended to convert an application from one IBB system to another in order to run on other systems. In other words, this implies 'porting' an application to other systems. Such porting can be achieved when a functional equivalent exists among the considered systems. In the IBB systems considered in Recommendation ITU-R BT.2075, many equivalent functions are found. For example, the application launch process for service associated applications in each system is very similar. For functionalities which exist only in specific IBB system(s), it is necessary to use other approaches to achieve the same functionalities on other systems. However, functionalities covered by this approach can be executed on the existing IBB receivers, which is very cost effective on the receiver side.

**c) Server-side/Cloud rendering**

Considering the rapid development of broadband technologies, the installation of 'a virtual receiver' on the server-side has recently become feasible. Figure 3-2 shows how 'a virtual receiver' works.

FIGURE 3-2

**Server-side/cloud rendering**




A virtual receiver receives a broadcast signal and executes the IBB applications. The virtual receiver is located in the server(s) which can be in the cloud and does not have a directly connected screen. The virtual receiver decodes audio visual broadcast signals and renders graphics as instructed by the IBB applications on its video memory. The receiver transfers the rendered images in the video memory to a screen device over a broadband network. The receiver also responds to commands by a remote controller, which communicates via a device for the screen or via the broadband network.

In this approach, a virtual receiver is a receiver that is compliant with a specific IBB system. Applications for such a specific IBB system run on this virtual receiver. When applications for other IBB systems need to be run, the corresponding virtual receivers should be used.

The result of the rendering, which is an audio and a composition of a video image and graphics, is presented on a screen over the broadband network in order to minimize the role of a local receiver. Since the currently available bandwidth of a broadband network does not reach that of the uncompressed video images of HDTV, re-compression of images may be required to deliver the images from the virtual receiver to the screen device. Such re-compression will cause a degradation of picture quality and a delayed response to the users' manipulation on a remote controller. Further, a special application to make an IBB receiver work as a screen device on each IBB system is required. Therefore, while this approach assures the execution of IBB applications, the user experience strongly depends on the situation of the broadband network.

## 2.2 Commonality of application type

In Recommendation ITU-R BT.2053, two major types of applications are defined, namely a service associated application and a stand-alone application. In the stand-alone application, there are various requirements in each IBB system. These requirements are mostly achieved by the middleware of the IBB systems, which leads to differences across the IBB systems. On the other hand, behavioural models for service associated applications are similar to all the IBB systems in Recommendation ITU-R BT.2075.

For example, when comparing Hybridcast, HbbTV 2.0, TOPSmedia and Ginga, they support service-associated IBB applications and stand-alone IBB applications. Table 3-1 shows their application types for these kinds of applications.

TABLE 3-1

**Application types**

| | Service-associated IBB application | Stand-alone IBB application |
|---|---|---|
| Hybridcast | Broadcast-oriented managed application | Non-broadcast-oriented managed application |
| HbbTV 2.0 | Broadcast-related application | Broadcast-independent application |
| TOPSmedia | Broadcast-activated application | Broadcast-inactivated application |
| Ginga | Associated local client | Stand-alone local client |

There are differences in the defined application behaviour in each system for stand-alone IBB applications. For Hybridcast, it is supposed that non-broadcast-oriented managed applications are primarily launched by 'application launcher' installed in a receiver while broadcast-independent applications in HbbTV 2.0 are primarily launched by broadcast-related applications or other broadcast-independent applications. In HbbTV 2.0, 'application launcher' in a receiver is optional. For TOPSmedia, the broadcast-inactivated applications can be launched by broadcast-activated applications or other broadcast-inactivated applications. And it also supports 'application launcher' in the case of pre-installed applications also known as store applications. In Ginga, stand-alone clients may be authorized (or not) to access broadcast resources. These applications may be launched from any other application by calling the Ginga WebServices Remote APIs or from the 'Application Catalogue UI'.

These systems behave similarly regarding the life cycle of stand-alone IBB applications which is not controlled by broadcasters. A non-broadcast-oriented managed application in Hybridcast and a broadcast-independent authorized application in Ginga are allowed to access to broadcast resources when a broadcaster whose offering resource is about to be accessed permits the application to do so. That is, each broadcaster can transmit access permission information to each application of these types. Broadcast-independent application and broadcast-inactivated application in HbbTV 2.0 and TOPSmedia are not allowed to access to broadcast resources. However, a broadcast-independent application can be transitioned to a broadcast-related application. Once a broadcast-independent application transitions to a broadcast-related application, it is allowed to access to broadcast resources.

In HbbTV 2.0, an application is allowed to transition between broadcast-related application and broadcast-independent application in both directions under the control of broadcasters. In Hybridcast, a broadcast-oriented managed application can transition to a non-broadcast-oriented application only. That is, a non-broadcast-oriented managed application is not allowed to transition to a broadcast-oriented managed application. It is considered that a reason for this difference between Hybridcast and HbbTV 2.0 is a difference of capability of parallel execution of multiple applications. Hybridcast is designed to execute multiple non-broadcast-oriented managed applications in addition to a broadcast-oriented managed application simultaneously in the same receiver.

Meanwhile, TOPSmedia only allows the transition from broadcast-activated application to broadcast-inactivated application under the control of broadcasters and it also support one application execution at a given time like HbbTV 2.0.

In Ginga, applications may be allowed to transition to another type in the same lifecycle, depending on the implementation. Additionally, an application can freely launch applications of another type.

Considering the difference of behaviour of stand-alone applications of each system, focusing on service-associated IBB applications is a reasonable approach for further analysis.

### 2.3 Practical approach

On the basis of the analysis described in §§ 2.1 and 2.2, it is considered that the application conversion, described in § 3, for service associated IBB applications is the most practical approach for the harmonization of IBB systems.

## 3 Comparison of application environment of various IBB systems

This chapter describes a comparison of the application environments of IBB systems in both aspects of syntax and behaviour. Section 3.1 describes the comparison in terms of syntactical aspects. Section 3.2 describes comparisons of life-cycle aspect. As to the application conversion, the differences of behaviour among each IBB systems should be carefully considered.

### 3.1 Comparison of additional objects, properties, and methods

As Hybridcast, HbbTV 2.0, TOPSmedia and Ginga-HTML5 are based on HTML5, functions provided by APIs, elements, attributes, etc. defined in HTML5 and other related Recommendations of World Wide Web Consortium are considered common between these systems.

On the other hand, each system defines additional objects and their properties and methods[9], elements[10], attributes, etc. to provide functionalities needed for IBB systems. Those additions are not common between the systems. However, considering similarity of behaviours of service-associated applications in these systems, there may be similarity or equivalency in those additions in the systems. Table 3-2 shows equivalent additional objects, properties and methods that have been identified by detailed analysis of the systems. Note that Table 3-2 does not consider combination of additional methods to provide the equivalent function among the systems and that properties and methods specific to a system in the object are not listed in Table 3-2. For example, identification of StreamEvent is incompatible between the systems because it is strongly tied to multiplexing scheme, i.e. ARIB data carousel and DVB object carousel.

In the case of Ginga-HTML5, the needed IBB functionalities are enabled by remote APIs based on the REST architectural style, provided by ‘Ginga Common Core Web Services’ (ABNT NBR 15606-11). Therefore, instead of JavaScript objects’ methods and/or properties, Ginga-HTML5 applications access functions and data via HTTP requests and get the returned values mostly in JSON format. Ginga-CC WebServices provides these remote HTTP APIs to broadcaster-authorized Ginga and non-Ginga applications. In this way, any application running on devices in the home network (TV, SmartTV, SmartPhone etc.) may be authorized to be part of the IBB experience. Ginga-NCL (ABNT NBR 15606-2) applications may also access this remote API, if needed, although most IBB functionalities can also be found in its NCLua API.

If considering the application conversion, the way of harmonization should convert APIs from an IBB system to another one based on the equivalency described in Table 3-2.

---

[9] In this context, property and method are the terms used in object-oriented programming. Property represents data or event relevant to an object. Method is a function to give a behaviour of an object.

[10] In this context, the term ‘element’ is a technical term used in XML. Elements can behave as containers to hold text, elements, attributes, media objects or all of them.

TABLE 3-2

**Equivalent additional objects, properties and methods**

| Hybridcast | | HbbTV 2.0 | | TOPSmedia | | Ginga |
|---|---|---|---|---|---|---|
| **Object** | **Property or Method** | **Object** | **Property or Method** | **Object** | **Property or Method** | **REST API (HTTP request)** |
| ISDBResourceReference | original_network_id<br>transport_stream_id<br>service_id | Channel | onid<br>tsid<br>sid | Channel | number<br>type | http(s)://<host>/dtv/current-service GET<br>{ serviceContextId, serviceName, transportStreamId, originalNetworkId, serviceId } |
| List of available channels can be obtained by query to a server. | | ChannelList | getChannel() | ChannelList | item() | http(s)://<host>/dtv/service-list GET<br>{ { serviceContextId, serviceName, transportStreamId, originalNetworkId, serviceId }, ...} |
| NavigatorApplicatonManager | getOwnerApplication() | oipfApplicationManager | getOwnerApplication() | ApplicationManager | tvExt.application.appmgr | http(s)://<host>/dtv/<service-context-id>/apps GET<br>{ { appid, name, type, controlCode, state, entryPoint }, ... } |
| Application | replaceApplication()<br>destroyApplication() | oipfApplication | createApplication()<br>destroyApplication() | ApplicationManager | createApplication()<br>destroyApplication() | http(s)://<host>/dtv/current-service/apps/<appid> POST<br>{ action: start/stop } |
| | visibility and/or display of style property of visible elements | Application | show()<br>hide() | ApplicationManager | showApplication()<br>hideApplication() | |

TABLE 3-2 (*continued*)

| Hybridcast | | HbbTV 2.0 | | TOPSmedia | | Ginga |
|---|---|---|---|---|---|---|
| **Object** | **Property or Method** | **Object** | **Property or Method** | **Object** | **Property or Method** | **REST API (HTTP request)** |
| KeySet | RED<br>GREEN<br>YELLOW<br>BLUE<br>NAVIGATION<br>VCR<br>NUMERIC<br>value | KeySet | RED<br>GREEN<br>YELLOW<br>BLUE<br>NAVIGATION<br>VCR<br>NUMERIC<br>value | ApplicationManager | keySetValue<br>(it defines constant values: KEY_RED, KEY_GREEN, KEY_YELLOW, KEY_BLUE, KEY_NAVIGATION, KEY_VCR, KEY_NUMERIC, KEY_OTHER) | http://<host>/dtv/current-service/ginga/keyset GET<br>numeric_keys, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9<br>interactive_keys, colored_keys, RED, GREEN, YELLOW, BLUE, selection_keys, ENTER, BACK, EXIT, cursor_keys, CURSOR_LEFT, CURSOR_RIGHT, CURSOR_DOWN, CURSOR_UP |
| | setValue() | | setValue() | | setKeySet() | http://<host>/dtv/current-service/ginga/keyset POST |
| Capabilities | hasCapability() | oipfObjectFactory | isObjectSupported() | DeviceStatus | xmlCapabilities | http(s)://<host>/dtv/platform-capabilities GET |
| ReceiverDevice | tuneTo() | video/broadcast object | setChannel()<br>prevChannel()<br>nextChannel() | HTML5 video element | src | http(s)://<host>/dtv/<service-context-id> POST |
| | getCurrentEventInformation() | oipfApplicationPrivateData | currentChannel | ChannelManager | getCurrentChannel() | http(s)://<host>/dtv/current-service GET |
| | | ChannelConfig | currentChannel | | | |
| StreamEventTarget | addGeneralEventMessageListener() | video/broadcast object | addStreamEventListener() | Not support StreamEvent though the DSMCC<br>(it can be substituted by W3C WebSocket or Server-Sent event on the broadband network) | | http(s)://<host>/dtv/current-service/dsmcc/stream-events/<component-tag>/<carrousel-id> GET |
| | removeGeneralEventMessageListener() | | removeStreamEventListener() | | | http(s)://<host>/dtv/current-service/dsmcc/stream-events/<handle> DELETE |

TABLE 3-2 (*continued*)

| Hybridcast | | HbbTV 2.0 | | TOPSmedia | | Ginga |
|---|---|---|---|---|---|---|
| **Object** | **Property or Method** | **Object** | **Property or Method** | **Object** | **Property or Method** | **REST API (HTTP request)** |
| Broadcast audio/video object | enableFullScreen() | A/V Control Object | setFullScreen() | HTML5 video/audio element | requestFullScreen() | http(s)://<host>/dtv/current-service/<comp-tag> POST { action, pos, vol } |
| | | video/broadcast object | setFullScreen() | | | |
| | disableFullScreen() | A/V Control Object | setFullScreen() | | exitFullScreen() | |
| | | video/broadcast object | setFullScreen() | | | |
| | enableAudioMute() | A/V Control Object | setVolume() | | muted<br>volume | |
| | | video/broadcast object | setVolume() | | | |
| | disableAudioMute() | A/V Control Object | setVolume() | | | |
| | | video/broadcast object | setVolume() | | | |
| | isAudioMute() | video/broadcast object | getVolume() | | | http(s)://<host>/dtv/current-service/<comp-tag> GET |
| EITSchedule* | name<br>description<br>start_time<br>duration<br>service_id<br>transportstream_id<br>original_network_id | Programme | name<br>description<br>startTime<br>duration<br>channelID | Program | name<br>description<br>startTime<br>duration<br>channel<br>audioInfoArray<br>videoInfo<br>captionInfoArray | http(s)://<host>/dtv/<service-context-id>/info/epg GET<br><br>Current programme is identified in the 'present' field in the returned JSON structure |
| RecordedContentInformation** | start_time<br>duration | | | | | |
| ReceiverDevice** | getRecordedContenInformation() | | selectComponent()<br>unselectComponent()<br>getComponents() | | | |

TABLE 3-2 (*end*)

| Hybridcast | | HbbTV 2.0 | | TOPSmedia | | Ginga |
|---|---|---|---|---|---|---|
| **Object** | **Property or Method** | **Object** | **Property or Method** | **Object** | **Property or Method** | **REST API (HTTP request)** |
| EITSearchManager | onEITScheduleUpdate()<br>onEITSearch() | oipfSearchManager | onMetadataUpdate()<br>onMetadataSearch() | Not support metadata search and query mechanism | | http(s)://<host>/dtv/<service-context-id>/info/epg GET<br>or<br>http(s)://<host>/dtv/all_services/info/epg GET<br><br>Metadata search must be implemented by the application, based on the returned JSON structure |
| | createSearch() | | createSearch() | | | |
| EITSearch | result<br>setQuery()<br>createQuery() | MetadataSearch | result<br>setQuery()<br>createQuery() | | | |
| Query | and()<br>not() | Query | and()<br>not() | | | |
| SearchResults | length<br>offset<br>totalSize | SearchResults | length<br>offset<br>totalSize | | | |
| | item()<br>getResults()<br>abort() | | item()<br>getResults()<br>abort() | | | |

Notes:

* Since this object is a part of the results of EITSearch, no constructor is defined. However, information contained in this object is equivalent to corresponding object(s) of other systems.

** These objects are applicable only to recorded content.

## 3.2 Comparison of application life-cycle aspects

### 3.2.1 Application life-cycle

Application life-cycle deals with the process that takes place from an IBB application's start to its end. To manage the application life-cycle exactly, IBB systems define each application model and the methods for starting and stopping the applications. Table 3-3 shows ways to launch the specific type's application.

In addition, HbbTV 2.0 and TOPSmedia support single application execution at any given time. Thus, if an application creates a new application, the former application is terminated and then the latter application is launched. Even though it seems to be a transition from one application to another from the user's point of view, it is actually different from an application transition. That is, the sequence of the application terminates and launches between two different applications.

TABLE 3-3

**Launch of the application**

| | HbbTV 2.0 | Hybridcast | TOPSmedia | Ginga |
|---|---|---|---|---|
| Launch the service-associated IBB application | – AUTOSTART application by broadcaster's signal (AIT)<br>– createApplication() using 'dvb:' URL scheme listed in AIT | – AUTOSTART application by broadcaster's signal (AIT)<br>– startAITControlledAPP() | – AUTOSTART application by broadcaster's signal (AIT)<br>– createApplication() using 'tvapp:' URL scheme listed in AIT<br>– User selection of installed(store) application having 'activated' attribute | – AUTOSTART application by broadcaster's signal (AIT)<br>– Stream Event (startDocument() editing command)<br>– User selection among the available applications (installed or signalled) via Application Catalogue UI<br>– Via Ginga CC WebServices API for controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST<br>– Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST |
| Launch the stand-alone IBB application | – createApplication() using HTTP/HPPTS or XML AIT<br>– By terminal specific application (e.g., TV portal) | – startAITControlledAPP()<br>– replaceApplication() from service-associated IBB application or stand-alone IBB application<br>– By terminal specific application (application launcher) | – createApplication() using HTTP/HPPTS<br>– User selection of installed(store) application having 'inactivated' attribute | – Stream Event (startDocument() editing command)<br>– User selection among the available applications (installed or signalled) via Application Catalogue UI<br>– Via Ginga CC WebServices API for controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST<br>– Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST |

### 3.2.2 Application signalling

To control the life-cycle of service-associated applications, the broadcaster can send the application signalling information through the current broadcast channel. HbbTV 2.0 and TOPSmedia use the AIT (application information table), and they refer to ETSI TS 102 809 according to each profile definition. Hybridcast uses the AIT defined in ARIB STD-B24. Ginga also uses the AIT, with an extended set of control codes. Table 3-4 shows the application control code of the systems.

According to the current AIT information sent by the broadcaster, the current application on the receiver, such as launch (autostart), keep (present), terminate (kill), etc., can be controlled by the broadcaster. However, in the AIT definition of HbbTV 2.0 and TOPSmedia, there is no information about the application transition that is described in the next paragraph. Thus, an application is not allowed to transit between two application types under the control of the broadcaster's signalling.

TABLE 3-4

**Application control code**

| | **HbbTV 2.0** | **Hybridcast** | **TOPSmedia** | **Ginga** |
|---|---|---|---|---|
| Application control code | 0x01 AUTOSTART<br>0x02 PRESENT<br>0x04 KILL<br>0x07 DISABLED | 0x01 AUTOSTART<br>0x02 PRESENT<br>0x04 KILL<br>0x05 PREFETCH | 0x01 AUTOSTART<br>0x02 PRESENT<br>0x04 KILL | 0x01 AUTOSTART<br>0x02 PRESENT<br>0x03 DESTROY<br>0x04 KILL<br>0x05 PREFETCH<br>0x06 REMOTE<br>0x07 UNBOUND<br>0x08 STORE<br>0x09 STORED_AUTOSTART<br>0x0A STORED_PRESENT<br>0x0B STORED_REMOVE |

### 3.2.3 Application transition

Application transition occurs when the application changes its current type while keeping the current application life-cycle. In other words, the change of the application type should happen in one application without any application termination and launch.

Table 3-5 shows the comparison of the application transition for each of the systems. HbbTV 2.0 supports this with some methods for application transition between the broadcast related application and the broadcast independent application in both directions, whereas Hybridcast and TOPSmedia only support the application transition in one direction from service-associated IBB application to the stand-alone IBB application.

TABLE 3-5

**Application transition**

| | HbbTV 2.0 | Hybridcast | TOPSmedia | Ginga |
|---|---|---|---|---|
| From service-associated to stand-alone | setChannel() as null for its channel argument | replaceApplication() using AIT and URL | createApplication() as an empty parameter | Via Ginga CC WebServices API for running a target application via a deep link http(s)://<host>/dtv/target-apps POST |
| From stand-alone to service-associated | Service selection under the conditions of § 6.2.2.6.1 | Not supported | Not supported | Via Ginga CC WebServices API for controlling an interactive application http(s)://<host>/dtv/current-service/apps/<appid> POST |

## 4 Harmonization of second screen handling

### 4.1 Communication protocols used between devices

Second screen capability can be provided in several ways. For example, relay servers for communication between devices (a TV set and second screen devices) enable a second screen IBB service without the consideration of harmonization of IBB systems because the service can be implemented only at application level. Therefore, all the functionalities required for IBB systems involve offering communication capability between the devices and servers, which the IBB systems provide according to their specification. Further, when communicating between devices directly without a relay server, harmonization of second screen handling is required.

In the case of direct communication between devices, it is required to use two types of protocols between a TV set and the second screen devices in general. Each protocol is used in each of the phases of device discovery and communication in order to enable the services involving the second screen capability to work correctly.

Device discovery is the required action to find out the partner device for the second screen service. Various home use environments exist – there are several combinations of manufacturers, types of devices, and network installations. Applications for a second screen service require seeking a partner device in various environments. Therefore, the device discovery phase is the first phase to establish a sufficient environment for the applications. Once the application finds an appropriate partner device successfully, communications between the application and/or IBB system capabilities enable to launch and maintain the service.

These two phases are required in both cases where the protocols are proprietary or standardized. When using the proprietary protocols, the IBB system software installed in both devices can be easily integrated with the proprietary services such as those provided by TV manufacturers. In addition, because a single entity such as a TV manufacturer implements the software for both devices less compatibility and/or paring problems can be expected. However, the user interface of the software may vary depending on the implementers and thus it may be difficult to provide the same user experience from the viewpoint of IBB service providers.

When using standardized protocols, multiple implementers will develop the software for functions of device discovery and communications for applications in the same manner. Thus, there is an open environment to provide second screen services and the IBB service providers will obtain more unified

user interfaces than in the case of proprietary protocols, provided the conformance of the software is maintained.

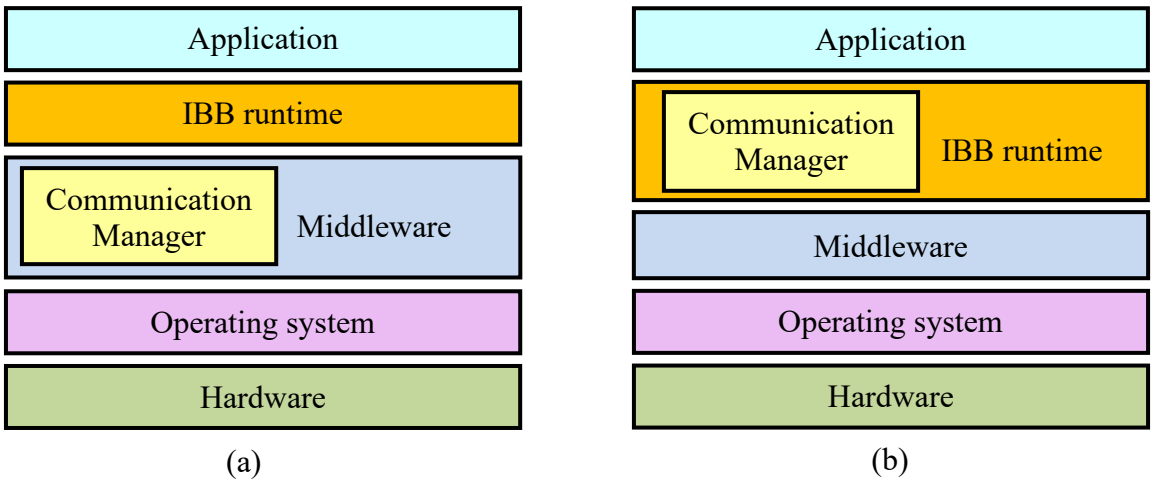
From an API perspective, the proprietary protocols require relatively high-level APIs to encapsulate the differences in the behaviours of the IBB system software based on the differences in the protocols. The standardized protocols allow lower-level APIs than proprietary protocols in certain cases.

## 4.2 Communication management

Second screen IBB services are built on top of the capability of the communication between devices. The management of the communication is relatively important to launch and maintain the services. The allocation of the communication manager in the software stack is influential of the flexibility in determining the services.

FIGURE 3-3

**Possible location of the communication manager**




In Fig. 3-3, the case (a), which is applicable to Hybridcast and Ginga, illustrates the case where the communication manager is located in the middleware layer. In this case, an IBB application is isolated from the communication manager and hence the APIs for communication management are considered as high level APIs. Further, the communication management is primarily controlled by the middleware and it is transparent for the application. Instead, this location of the communication manager allows the manager to work independently of the applications. Thus, the software in the companion device can communicate with lower layer software in the TV set, which allows the software in the companion device to have an opportunity to control the TV sets regardless of the state of execution of the application such as on and off of power of the TV set, or launch control of the application in the TV set.

In the case (b), which is applicable to HbbTV and TOPSmedia, the communication manager is located in the IBB runtime layer. This allocation is typically intended to provide APIs to control the communication manager via the application directly. Further, the application is in charge to control all the behaviours of the second screen services including device discovery. This structure provides flexibility to the application for services such as the selection of a partner device by the application. In contrast, the software in the companion device communicates with the manager in the IBB runtime, and subsequently, then difficulty may arise for the software to communicate with the lower-level functions in the TV set.

Considering the harmonization of the applications, certain supplemental application programme code may be required to achieve equivalent behaviours between the cases that may or may not contain the APIs for device discovery and pairing of the devices.