



UNIVERSIDAD POLITÉCNICA DE CARTAGENA
E.T.S. Ingeniería de Telecomunicación



DESARROLLO DE UN SERVICIO DE TELEVISIÓN INTERACTIVA HbbTV SEGÚN EL ESTÁNDAR ETSI TS 102 796 v1.1.1 (JUN 2010)

Titulación: Ingeniero Técnico de Telecomunicación, esp. Telemática

Alumno: José Daniel Gambín Tomasi

Director: Juan Luis Pedreño Molina

Codirector: Diego Lorenzo Martín (RTRM)

Cartagena, 16 de Julio de 2012



Título del PFC

DESARROLLO DE UN SERVICIO DE TELEVISIÓN INTERACTIVA HbbTV
SEGÚN EL ESTÁNDAR ETSI TS 102 796 v1.1.1 (JUN 2010)

Autor

José Daniel Gambín Tomasi (*daniel.tomasi@gmail.com*)

Director

Juan Luis Pedreño Molina (*juan.pmolina@upct.es*)

Codirector

Diego Lorenzo Martín (*diego.lorenzo@rtrm.es*)

Titulación

Ingeniero Técnico de Telecomunicación especialidad Telemática

Departamento

Tecnologías de la Información y las Comunicaciones

Defensa

En Cartagena, Julio de 2012

Resumen

La televisión híbrida es el último avance en el campo de la televisión interactiva. Se fundamenta en la unión de la televisión tradicional por aire/cable/satélite (Broadcast) y la distribución de contenidos por redes IP (Broadband). De esta forma, ambos mundos convergen en un mismo punto, el televisor.

El planteamiento inicial del proyecto es comprobar las posibilidades de la especificación HBBTV para exponer las características más interesantes de este nuevo estándar.

El objetivo principal de este proyecto es el desarrollo de un sistema que sirva para comprobar las características de este nuevo estándar.

Se pretende que durante la emisión de un programa de televisión pueda accederse a contenidos extra ofrecidos por el canal Broadband haciendo uso de una aplicación HBBTV.

Descriptor: Televisión interactiva, HbbTV, broadcasting, televisión híbrida, video on demand.



BLOQUE 1 – Introducción a la Televisión interactiva

1. Introducción a la Televisión interactiva	9
1.1 ¿Qué es la televisión interactiva?	9
1.2 TVi	9
1.2.1 MHEG	10
1.2.2 DAVIC	12
1.2.3 MHP	12
1.3 TVi híbrida	16
1.3.1 MHP 1.3 (GEM1.3)	16
1.3.2 MHEG-IC	16
1.3.3 HbbTV	17
1.3.4 DTG UK CTV	19
1.4 Web TV, IPTV, Hybrid TV y OTT	20
1.5 Comparativa de estándares	22
2. HbbTV	23
2.1 ¿Por que se crea este nuevo estándar?	23
2.2 Canal Broadcast y canal Broadband	29
2.3 Tipos de servicios que posibilita HbbTV	30
2.3.1 BC-related	30
2.3.2 BC-independent	30
2.3.3 Ejemplos de servicios	33
2.4 Tecnologías implicadas	37
2.4.1 CE-HTML	37
2.4.2 OIPF (DAE y Formatos)	37
2.4.3 Señalización y Transporte	38
2.5 Estudios	39
2.5.1 Estudios de mercado	39
2.5.2 Estudios de usabilidad	40
2.6 Sistemas o plataformas alternativas a HbbTV	41
2.6.1 Fabricantes de televisores, STBs, etc.....	41
2.6.2 Apple TV	42
2.6.3 GoogleTV	43
2.6.4 Youview	44
2.6.5 Calendario	46
2.7 Despliegue (junio 2012)	46
3. HbbTV en España	47



BLOQUE 2 – Desarrollo técnico del estándar HbbTV

4. Desarrollo técnico del estándar HbbTV	51
4.1 Introducción	51
4.2 Plataforma HbbTV	52
4.3 Visión general	53
4.4 Arquitectura de la plataforma HbbTV	54
4.4.1 Introducción	54
4.4.2 Sistema	54
4.4.3 Terminal	55
4.4.4 Capacidades mínimas y opcionales del terminal	56
4.5 Fundamentos tecnológicos del estándar HbbTV	57
4.5.1 CEA-2014 A	57
4.5.1.1 XHTML 1.0 Strict o Transitional	62
4.5.1.2 DOM 2.0	63
4.5.1.3 XMLHttpRequest	63
4.5.1.4 CSS TV Profile 1.0	64
4.5.2 OIPF	65
4.5.2.1 OIPF DAE	66
4.5.2.2 OIPF Media Formats	70
4.5.2.2.1 Video	70
4.5.2.2.2 Audio	73
4.5.2.2.3 Contenedores	75
4.5.2.2.4 Subtítulos	78
4.5.2.2.5 Resumen de formatos soportados en HbbTV	78
4.5.3 ETSI TS 102 809 – Señalización de aplicaciones	79
4.5.3.1 Señalización AIT	82
4.5.3.2 DSM-CC	85
4.5.3.2.1 Stream events	86
4.6 Protocolos	88
4.6.1 Streaming	88
4.6.1.1 HTTP	88
4.6.1.2 RTSP/RTP (+RTSP)	90
4.6.2 Descarga de contenidos (+DL)	92
4.6.3 Transporte de aplicaciones	93
4.6.4 Cabeceras User-Agent para el navegador HbbTV	94



4.7 Aplicaciones y servicios	95
4.7.1 Introducción	95
4.7.2 Ciclo de vida de la aplicación	96
4.7.2.1 Inicio y fin de las aplicaciones	96
4.7.2.2 Comportamiento de la aplicación al sintonizar un canal	96
4.7.2.3 Comportamiento mientras se permanece en el canal	99
4.7.2.4 Señalización simultánea de aplicaciones en bcast y bband	101
4.7.2.5 Aplicaciones bc-independent	102
4.7.3 Ejemplo del ciclo de vida de una aplicación	103
4.8 Seguridad	104
4.9 Capacidades y funciones del terminal	105
4.9.1 Capacidades mínimas	105
4.9.1.1 Acciones de usuario	106
4.9.2 Funciones del terminal	107
4.9.3 Consulta de capacidades y cadenas opcionales	107
4.10 Evolución del estándar	110
4.10.1 HbbTV 1.5 (futuro ETSI TS 102 798 v1.2.1)	110
4.10.1.1 MPEG DASH (Dynamic Adaptive Streaming over HTTP)	111
4.10.2 HbbTV 2.0	114

BLOQUE 3 - Desarrollo práctico de la especificación

5. Desarrollo práctico de la especificación HbbTV	115
5.1 Plan de desarrollo	115
5.2 Piloto	115
5.2.1 Aplicación básica	115
5.2.2 Ingesta de contenidos	117
5.2.3 Señalización	119
5.2.4 Conclusiones fase piloto	119
5.3 Arquitectura definitiva	120
5.3.1 Cabecera de red	120
5.3.2 Distribución de contenidos	122
5.3.3 Terminal HbbTV del usuario	123
5.4 Desarrollo de la aplicación definitiva	126
5.4.1 Consideraciones en cuanto al diseño de la aplicación	126
5.4.2 Entorno de desarrollo de las aplicaciones	128
5.4.3 Presentación de la aplicación final	130
5.5 Ejemplos de otros desarrollos y sistemas	133



6. Glosario	137
6.1 Definiciones	137
6.2 Terminología	140
7. Bibliografía	143
Anexo A – Estructura básica de una aplicación HbbTV	147
Anexo B – Funciones básicas para aplicaciones HbbTV	153



PRÓLOGO

Este proyecto nace por el interés de la dirección técnica de RTRM de estudiar nuevas tecnologías de televisión interactiva (TVi) que permitan sustituir los servicios de MHP que se venían prestando.

RTRM abandonó en 2010 los servicios de TVi que venía ofreciendo mediante el uso de la tecnología MHP. Esta tecnología no se abandonó porque no se creyera en ella, es una tecnología válida que ha evolucionado mucho y ha sabido adaptarse a los cambios y nuevas oportunidades del mercado, como explicamos en el primer capítulo de este proyecto. Pero comenzaron a haber reclamaciones de patentes y el mercado reaccionó dejando de fabricar terminales lo que llevó a que radiodifusores y desarrolladores de contenidos dejaran de producir contenidos para esta tecnología. Hay que recordar que MHP fue una iniciativa europea asumida a nivel nacional que contaba con el apoyo del gobierno a través de la Secretaría de Estado de Telecomunicaciones y Sociedad de la Información (SETSI).

En paralelo al declive de MHP, comienzan a aparecer en las tiendas las primeras televisiones interactivas denominadas InternetTV o SmartTV. Las condiciones para desarrollar contenidos para estas plataformas no nos convencen, por dos motivos: el económico y el práctico. Debido a que cada fabricante tiene su propia plataforma hay que realizar un desarrollo individual por cada marca, algo que económicamente es inviable. Y luego, en la práctica, la ubicación de la aplicación dentro del portal es una decisión del fabricante (acuerdos comerciales aparte) junto a otros condicionantes.

También se valoraron las plataformas de operador (Imagenio, ONO, Abertis) y de proveedores de contenidos (GoogleTV, AppleTV). Pero los inconvenientes eran que el despliegue de estas plataformas no es mayoritario y que suelen usar plataformas específicas, verticales y no abiertas, heredando los mismos inconvenientes que encontramos en las plataformas de los fabricantes de televisores.

Por la misma época en la que aparecen las televisiones híbridas (son híbridas porque disponen del canal de banda ancha por internet además del canal de radiodifusión) en junio de 2010 se publica un estándar que es resultado de una iniciativa francesa y alemana para el desarrollo de un mercado horizontal para la TVi. Este estándar es el ETSI TS 102 796 v1.1.1, también conocido como estándar HbbTV. En la dirección técnica de RTRM nos parece interesante este nuevo estándar horizontal frente a las opciones de mercado vertical propuestas por los fabricantes con las SmartTV o las plataformas de operador. Es un estándar abierto y horizontal como también lo era MHP pero no sometido a derechos por uso. En el primer capítulo del documento hacemos un repaso por los distintos estándares abiertos y horizontales que existen en la actualidad.

Decidimos estudiar el estándar HbbTV a fondo para valorar si era apropiado para sustituir nuestros servicios interactivos con tecnología MHP, además de una buena apuesta de futuro. El estudio del estándar y la valoración acerca de su conveniencia o no se corresponde con la primera parte de este PFC. Visto que HbbTV resultaba una tecnología conveniente y asequible económicamente además de viable en términos de complejidad de implantación, decidimos apostar por ella. Teniendo en cuenta que para julio de 2012 había un cambio de contrato para los servicios de transporte de la señal y la cabecera de red decidimos incorporar el soporte para HbbTV en el nuevo contrato.



El plan elaborado en 2011 por la dirección técnica para incorporar la tecnología HbbTV en las emisiones de 7RM fue el siguiente:

- 1 – Estudiar la conveniencia de llevar a cabo desarrollos para la plataforma HbbTV.
- 2 – Participar en el grupo de radiodifusores para el estudio de la televisión interactiva.
- 3 – Realización de un piloto.
- 4 – Lanzamiento de la aplicación definitiva.

Creímos que los puntos 1, 2, 3 y parte del 4 podían ser objeto del PFC, por lo que me responsabilizaron a mi del estudio y desarrollo de la parte interactiva.

Este encargo implicaba realizar las siguientes tareas:

- Un estudio de tecnologías existentes para el desarrollo de aplicaciones de TVi.
- Un estudio de en qué consiste HbbTV.
- Valorar la conveniencia del estándar.
- Participación en el grupo de radiodifusores para el estudio de la televisión híbrida.
- Coordinar el desarrollo de un piloto.
- Pruebas.
- Propuesta del alcance y funcionalidad de la aplicación definitiva.

Todo esto es lo que se contempla en este PFC.

El PFC está estructurado en tres bloques:

Bloque 1 – Introducción a la TVi (puntos 1 al 3)

En el punto 1 hacemos una introducción a la televisión interactiva. Exponemos los fundamentos tecnológicos de las distintas tecnologías y plataformas abiertas existentes. Hablamos acerca de su despliegue y listamos sus características más significativas. Tras ello introducimos la televisión híbrida y los estándares de TVi que se han desarrollado con ella. Finalmente hacemos una tabla comparativa entre todas las tecnologías adaptadas a la televisión híbrida, listando sus ventajas e inconvenientes.

En el punto 2, nos dedicamos exclusivamente a analizar las características que constituyen el estándar HbbTV, pues es considerado el más ventajoso para el desarrollo de este proyecto. También analizamos las plataformas alternativas a HbbTV y finalmente ponemos sobre un mapa el actual despliegue que tiene esta tecnología en Europa.

En el punto 3 hacemos el seguimiento de como se está desarrollando la introducción de la tecnología HbbTV en España. Hablamos de anteriores iniciativas como MHP, de las distintas partes implicadas en el proceso, los problemas que han surgido, los documentos ya generados en el proceso y los que actualmente a fecha de este proyecto aún están en desarrollo y en los que estamos participando como parte activa.



Bloque 2 – Desarrollo técnico del estándar HbbTV (punto 4)

En este segundo capítulo entramos en profundidad en los fundamentos tecnológicos del estándar HbbTV.

En el punto 4 listamos las tecnologías que componen HbbTV y enumeramos todas las funcionalidades y características de las que dotan al estándar. Repasamos los pilares que componen el estándar y diferenciamos que aporta cada uno de ellos, analizamos la funcionalidad ofrecida por el canal broadcast y la ofrecida por el canal broadband además de los protocolos y tipo de aplicaciones implicados. También vemos como es el ciclo de vida que tienen las aplicaciones según su tipo y sus esquemas de seguridad. Tras todo esto vemos cuales son los requisitos mínimos que exige el estándar para su aplicación en un terminal.

Al final del punto detallamos las funciones que aportan las nuevas versiones del estándar ya publicadas y adelantamos la probable fecha de lanzamiento de la siguiente versión aún no publicada junto con las posibles novedades que puede incluir.

Bloque 3 – Desarrollo práctico del estándar HbbTV (punto 5)

En el último capítulo del proyecto es donde exponemos los trabajos que se han realizado para poner en funcionamiento un servicio de televisión interactiva HbbTV en la cadena de televisión autonómica 7RM. Según el plan de desarrollo inicial, tras hacer el estudio de tecnologías y decidimos por HbbTV como la tecnología más adecuada para el desarrollo de servicios de televisión interactiva para la emisora autonómica 7RM, es el turno de desarrollar a nivel práctico la puesta en marcha del servicio. Para ello previamente ha habido que conocer todas las características del sistema y adaptar el entorno existente para que fuese posible incorporar HbbTV en el flujo de trabajo de 7RM. En este punto se explica como desarrolló la fase piloto y cual es el diseño de la arquitectura definitiva. Se presentan las aplicaciones piloto, y el esquema de bloques sobre el que se va a basar la aplicación definitiva.

Al final de punto se pone como ejemplo un desarrollo HbbTV muy complejo desarrollado por radiodifusores con más años de experiencia con esta tecnología.

ANEXOS

En los anexos incluimos el código de las aplicaciones básicas sobre las cuales estamos desarrollando las aplicaciones finales. En el anexo A tenemos la estructura básica de una aplicación HbbTV con los bloques obligatorios y en el Anexo B hemos extraído de nuestros desarrollos un pequeño grupo de funciones que permiten implementar distintas funcionalidades en nuestras aplicaciones.



BLOQUE 1 – Introducción a la Televisión interactiva

1.1 ¿Qué es la televisión interactiva?

Podemos definir la TV interactiva (TVi) como una forma de ver la televisión en la cual el usuario puede disfrutar de contenidos y servicios adicionales sumados a los contenidos de la televisión convencional, que tradicionalmente ha sido un medio de comunicación unidireccional. Estos contenidos y servicios adicionales venían siendo ofrecidos por el radiodifusor, pero hoy día también pueden ser ofrecidos por los fabricantes de los terminales o por terceros. El verdadero valor de la TVi reside en darle al usuario la oportunidad de participar y de interactuar con los contenidos que está recibiendo.

El grado de interactividad dependerá de las posibilidades que ofrezcan esos contenidos adicionales que hacen que la TV sea interactiva y de las características técnicas del sistema y del terminal del usuario. Podemos ir desde un nivel muy básico, en el cual la interactividad es solo local porque no se hace uso de un *canal de retorno*. En este caso el radiodifusor envía el contenido adicional y el usuario accede a él sin que ninguna de sus acciones sea comunicada de vuelta al radiodifusor (o proveedor de contenidos) para que ofrezca otro servicio o contenido como respuesta a esa acción. Un ejemplo de interactividad básica podría ser la información extendida que se ofrece de una película, con fotos, críticas, etc... El espectador ve en su televisor toda esa información pero no se lleva a cabo ninguna acción más allá de la simple consulta de la información en local.

Una experiencia realmente interactiva, con un grado de interactividad máximo supone que las acciones que haga el usuario relacionadas con el contenido que está recibiendo tengan como respuesta nuevos contenidos enviados por el radiodifusor (o proveedor de contenidos) en respuesta a las acciones del usuario. Para conseguir esto es necesario hacer uso del canal de retorno. El canal de retorno puede ser un SMS, una conexión telefónica con un módem o una conexión de banda ancha doméstica, por ejemplo.

Aunque la TVi ha tenido una incorporación lenta, afortunadamente, con la creciente competencia de canales en el espectro actual, está dando pasos firmes por consolidarse como una parte fundamental de la experiencia televisiva de los espectadores.

1.2 TVi

Desde la creación del Teletexto por la BBC (British Broadcasting Company) entre los años 1970-1976, los radiodifusores siempre han intentado desarrollar nuevas formas de llegar a los espectadores. La gran oportunidad se produce con la llegada de la televisión digital.

Al transformar la señal de televisión analógica en una señal digital (terrestre, por satélite o por cable) ya es posible incorporar fácilmente flujos de datos en la señal broadcast. Estos flujos de datos son recibidos descodificados e interpretados en los receptores. Lo que se hacía necesario eran estándares que definiesen los sistemas a utilizar (formatos, arquitectura, equipos, ...) para poder definir la parte interactiva de la televisión.

Aunque también existen plataformas privadas capaces de proveer de servicios interactivos, en este proyecto solo vamos a tratar de las **plataformas o estándares abiertos**. Hacemos un repaso según orden de aparición.



1.2.1 MHEG

MHEG es un estándar abierto de *middleware* diseñado específicamente para poder desarrollar servicios de TVi digital. La *ISO* lo desarrolló y estandarizó a mediados de los años 90 como parte del esfuerzo de la *DAVIC* por estandarizar, proporcionar interactividad y navegación para servicios de *video bajo demanda*.

MHEG es un estándar público que no está gravado con ningún tipo de derechos de uso.

Conceptualmente MHEG quiso ser a las aplicaciones multimedia interactivas lo que el HTML a las páginas web, es decir, un lenguaje común que permitiese la ejecución y presentación de contenidos en cualquier receptor.

Se crearon varias versiones:

- MHEG-1: Esta es la original. Usa la notación *ASN.1* (Abstract Syntax Notation One) para definir las aplicaciones multimedia basadas en objetos. *ASN.1* permite describir estructuras de datos complejas con una sintaxis no ligada a ningún lenguaje de programación. Después el compilador de *ASN.1* traduce a un lenguaje particular (C, C++, Java...)
- MHEG-3: Con esta versión se definió una máquina virtual estandarizada y la representación de las aplicaciones de tal forma que era posible portarlas entre las distintas plataformas (muy similar al funcionamiento de Java).
- **MHEG-5**: MHEG-1 y MHEG-3 eran conceptualmente algo complicados y la industria no estaba preparada para implementar las características que ofrecían. MHEG-5 fue publicado en 1997 como una versión simplificada de MHEG-1. Aun siendo una versión simplificada, en la práctica fue tratada como un estándar independiente.
- MHEG-6: En 1998 se creó MHEG-6, que no era más que MHEG-5 con soporte Java para el desarrollo de objetos de *scripting*. Se hizo creando un *API* de Java específico que era capaz de manejar objetos MHEG. MHEG-6 no tuvo popularidad pero fue la base de las especificaciones publicadas por la *DAVIC*.

De todas las versiones de MHEG, la única que ha tenido éxito es **MHEG-5**, siendo éste el estándar actual de TVi en el Reino Unido. Es una tecnología ya madura que cuenta con un mercado desarrollado de más de 27 millones de receptores (2008). El estándar ha sido adaptado al mercado del Reino Unido por el *DTG* dando lugar al "MHEG-5 UK Profile version 1.06" (actualmente ETSI ES 202 184 v2.1.1), que es la base del servicio de televisión digital terrestre Freeview. MHEG también es el estándar elegido en Freesat (Satélite, UK), Freeview Nueva Zelanda, Freeview Australia y TVB Hong Kong.

El sistema MHEG-5 es simple y fácil de implementar en receptores de bajo coste. El software de cliente se compone de un motor MHEG que interpreta las aplicaciones y presenta la información que le llega por el canal de radiodifusión en forma de *carrusel de objetos*. Los tipos de objetos que puede representar son video, sonido, listas, textos y gráficos. En cuanto a *eventos* acepta entradas del mando a distancia, de un contador interno, de *stream events* en el canal broadcast o de resultados de la propia aplicación.

Los perfiles broadcast de MHEG tienen un ciclo de vida de aplicación bastante simple ya que solo puede haber una aplicación en ejecución. Una aplicación MHEG puede lanzar otras pero al hacerlo la aplicación “lanzadora” se finaliza. También puede haber aplicaciones de tipo auto arranque (autostart), que se inician sin intervención del usuario al seleccionar un *servicio* (sintonizar un canal).

Las aplicaciones se cargan en el aparato a través de un carrusel de objetos que se recibe por el canal broadcast. La información a la que accede la aplicación es de tipo “incluido” cuando un texto o gráfico forma parte de la aplicación o “referenciado” cuando el contenido es releído del carrusel. Los datos cargados del carrusel pueden ir cambiando en determinados momentos lo que permite aplicaciones mucho más dinámicas.

Las aplicaciones MHEG en ejecución toman el control de la presentación y escalado del video y del sonido. Incluso permite sintonizar a otro canal o moverse entre distintos flujos de video dentro del mismo múltiplex. Esto podría permitir al radiodifusor mostrar más de una vista de cámara de un evento o cambiar a otro canal.

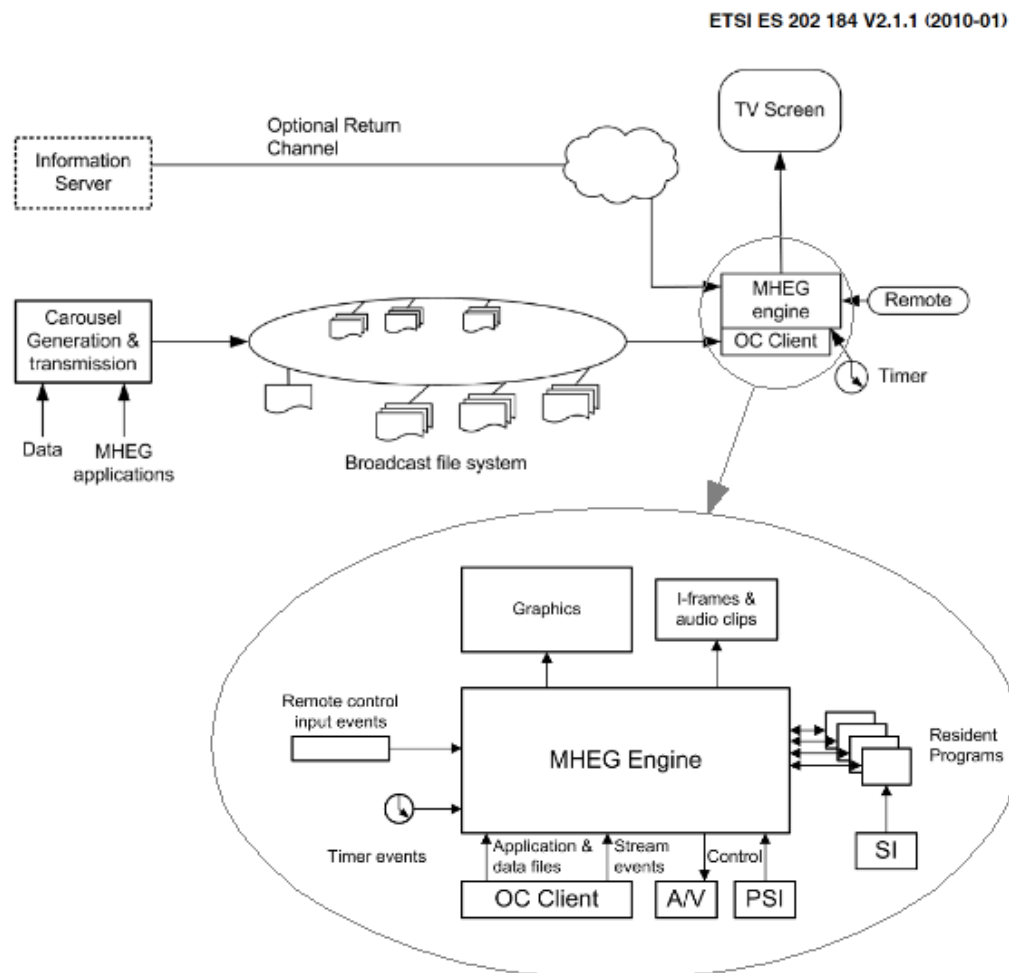


Figura 1.1 – Esquema de funcionamiento MHEG con detalle del motor de ejecución de la aplicación.



1.2.2 DAVIC

La *DAVIC* (Digital Audio Video Council) fue fundada en 1994 con el objetivo de promover el uso de servicios y aplicaciones audiovisuales mediante la creación de protocolos e interfaces abiertos que permitiesen la máxima interoperabilidad no solo geográfica sino también a nivel de aplicaciones, servicios e industrias. Fue una organización sin ánimo de lucro con sede en Suiza.

La *DAVIC* también publicó sus conjuntos de especificaciones. La última de ellas la *DAVIC 1.4.1* en 1999 (que se puede descargar de <http://docbox.etsi.org/Reference/DAVIC/>). La especificación 1.5 fue un conjunto de herramientas y definiciones adicionales que abrieron el camino a servicios audiovisuales basados en *IP*, entre ellos *TV Anytime*. En la actualidad el concepto de *TV Anytime* está resurgiendo en la *EBU*.

DAVIC tomó el estándar *MHEG-6* y lo amplió con una serie de nuevos APIs Java y soporte al Java Media Framework para controlar la reproducción multimedia. Aunque no era posible desarrollar una aplicación 100% Java con *DAVIC*, las APIs de Java que llevaba incluidos permitían controlar más elementos del receptor que cualquiera de los estándares anteriores.

1.2.3 MHP

MHP fue el primer estándar abierto de middleware basado completamente en Java. Las primeras versiones del estándar tuvieron que esperar a ser publicadas porque hacían referencia a la especificación *JavaTV* que se encontraba en estado de borrador. Cuando Sun Microsystems terminó la especificación, el estándar *MHP* se publicó en un corto periodo de tiempo.

MHP fue desarrollado por el proyecto *DVB* con la intención de solucionar el problema del mercado fragmentado y de las soluciones verticales que había entre los servicios ofrecidos por los radiodifusores del sector público y privado.

El *DVB* es un consorcio liderado por la industria con alrededor de 250 miembros entre los que se encuentran radiodifusores, fabricantes, operadores de red, desarrolladores de software, organismos reguladores, etc... distribuidos en más de 35 países y cuyo objetivo es el desarrollo de estándares técnicos abiertos para facilitar la distribución mundial de televisión digital y sus servicios de datos asociados.

Debido a la naturaleza del organismo que lo promovió, el objetivo principal de *MHP* era fomentar la construcción de mercados horizontales, donde hubiera una fuerte competición entre proveedores de servicios, operadores de red y fabricantes. El requisito principal era que hubiese interoperabilidad total entre las aplicaciones y las distintas implementaciones del estándar. Los radiodifusores necesitaban saber que sus aplicaciones iban a comportarse de una forma consistente y predecible en cualquier plataforma que cumpliera con el estándar.

Teóricamente *MHP* debía triunfar en mercados en los que no había tecnologías previas que hubiese que migrar o que hacer compatibles con la nueva. En el Reino Unido, donde si que ya había una tecnología en producción (*MHEG-5*), se recibió como un posible camino hacia el cual poder evolucionar.

En 1997 el *DVB* crea el grupo que debía definir los requisitos comerciales. Una de las primeras metas era asegurar que hubiese un acceso libre y no discriminatorio a *MHP*.



Además:

- MHP debía estar basado en estándares abiertos además de reutilizar los ya existentes del DVB
- MHP debía ser independiente del hardware y los sistemas operativos
- Se debía dar soporte para facilitar la migración de los servicios interactivos ya existentes hacia los nuevos
- Se debía definir de forma modular, lo que permitiría su futura evolución sin romper la retrocompatibilidad
- Se debía proporcionar suficiente información y baterías de test para posibilitar la autocertificación de la implementación hecha por el propio fabricante

MHP eligió Java como lenguaje de desarrollo porque en aquel momento, se intentó que fuera compatible con el resto de tecnologías existentes (MHEG-5, HTML, OpenTV, ...) pero era demasiado complicado. Pero se identificó un punto común y era que todas esas tecnologías pensaban en Java como el lenguaje para su evolución. También había otras razones para elegir Java:

- Los radiodifusores comenzaban ya a buscar la convergencia de la radiodifusión y de internet y Java fue un lenguaje diseñado desde el principio para entornos conectados en red. Incorpora soporte para protocolos de internet y tiene muchos mecanismos de seguridad para evitar aplicaciones maliciosas.
- La máquina virtual de Java no favorece a ningún hardware ni tecnología de implementación en particular.

Esas características han hecho que Java consiga extenderse ampliamente en una gran variedad de industrias, aunque en el momento de ser adoptado por el DVB para el estándar MHP era una tecnología inmadura en dispositivos de electrónica de consumo.

DVB inicialmente especificó 3 perfiles distintos para el estándar MHP:

1 – Enhanced Broadcast Profile (ETSI ES 201 812 MHP 1.0.x): Es el perfil básico del MHP. Pensado para entornos de usuario no conectados a ninguna red y donde las aplicaciones se descargan por el canal de broadcast. Al no ser un entorno conectado, **NO** hay canal de retorno. Las aplicaciones diseñadas para este perfil proporcionan interactividad básica local, p.ej una EPG.

2 – Interactive Broadcast Profile (ETSI ES 201 812 MHP 1.0.x): Toma como base al anterior añadiendo cualquiera de los canales de retorno bidireccionales que define el DVB (IP mediante modem p.ej). En este perfil se permite mandar datos de vuelta al radiodifusor y por tanto permite aplicaciones con interactividad real como pueden ser servicios de comercio electrónico y televoto.

3 – Internet Access Profile (ETSI TS 102 812 MHP 1.1.1): Es una ampliación del estándar anterior que permite aplicaciones comunes de internet residentes en el terminal: web, email, streaming,. Está pensado para que el radiodifusor pueda enlazar sus aplicaciones interactivas con sus aplicaciones de internet.

DVB también define un formato de contenidos opcional llamado **DVB-HTML**. Es una especificación de HTML muy concreta para ser usada en entornos de radiodifusión. Las aplicaciones están basadas en *XHTML* (HTML hecho con *XML*, más estricto) junto con otras tecnologías del mundo web: *CSS2*, *DOM*, *ECMAScript* (*JavaScript* estandarizado).

DVB-HTML es opcional en los perfiles Interactive Broadcast e Internet Broadcast.

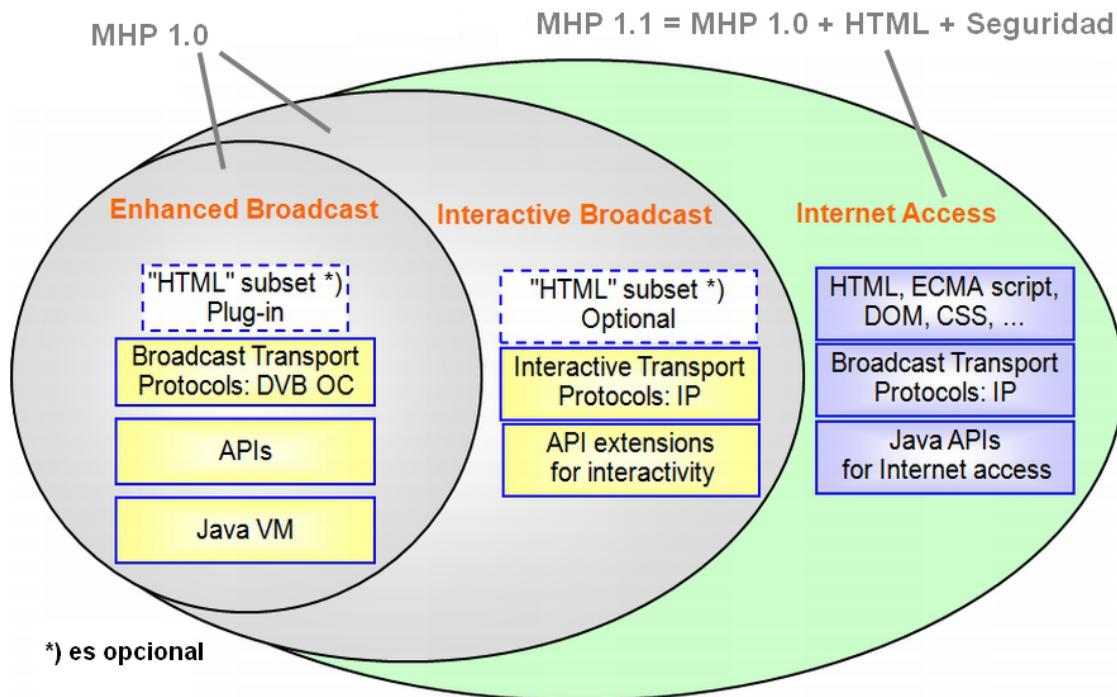


Figura 1.2 – Relación de las distintas versiones del estándar MHP entre sí.

En España, el 15 de Febrero de 2002 se firma un acuerdo de intenciones en el que administración, radiodifusores y fabricantes acuerdan usar MHP como tecnología estándar para el desarrollo de servicios interactivos en la *TDT*. Estas actuaciones fueron impulsadas por la administración central y autonómica exigiendo la adopción de MHP en los concursos convocados para las concesiones de los canales de TDT. El perfil más extendido es el MHP 1.0 Interactive Broadcast y es sobre el cual se hacen la mayor parte de las aplicaciones.

En 2009 la televisión autonómica de la Región de Murcia, 7RM, lanza un servicio MHP de televisión interactiva. Los aplicaciones interactivas incluidas eran:

- Cita médica: Permite solicitar cita previa con tu médico o pediatra en el centro de salud. Esta aplicación fue desarrollada en colaboración con el Servicio Murciano de Salud. Al ser una aplicación completamente interactiva era necesario disponer de un canal de retorno con conexión a internet. En su momento fue una aplicación pionera, recibiendo un premio y reconocimiento por su carácter innovador.
- EPG (Guía electrónica de programas): Para consultar los detalles de la programación de 7RM ampliada con fotos, sinopsis, más detalles...
- Ocio: También tenía disponibles juegos como las 4 en raya y el sudoku.
- En una segunda revisión también se incorporaron noticias y agenda cultural.



Algunas capturas de la aplicación general y de la aplicación de cita médica de 7RM:



Figura 1.3 – Distintas capturas de pantalla de la aplicación “Cita médica” de 7RM en el estándar MHP.

MHP finalmente fracasó en prácticamente todo el mundo excepto en Italia, donde a día de hoy sigue siendo ampliamente utilizado gracias a las subvenciones del gobierno para la compra de dispositivos compatibles. Estas ayudas fueron declaradas ilegales por la UE.

Los motivos para su fracaso fueron dos:

- El perfil más completo, el Internet Broadcast, era muy complicado de implementar.
- La industria perdió el interés debido a que el uso del estándar fue gravado finalmente con derechos de patentes. Debido a esto, el radiodifusor tenía que pagar por hacer aplicaciones y el fabricante pagaba por implementarlo en sus terminales.



1.3 TVi híbrida

Un terminal híbrido es un dispositivo (generalmente un TV) que se puede conectar en paralelo a dos redes ya que dispone de un sintonizador de televisión digital, de una conexión de banda ancha doméstica y de una plataforma de software que permite ejecutar servicios que unan a las dos redes. Una de esas redes es una red de radiodifusión (broadcast) por la cual le llegan las señales de los canales de televisión (y datos asociados) y la otra es una red de datos IP de banda ancha (broadband) por la cual le van a llegar aplicaciones y contenidos multimedia y no multimedia y por donde, además, se va a poder enviar hacia el radiodifusor las acciones y decisiones del usuario.

En España el declive de MHP se produce a partir del 2010. En otros países, esto había comenzado a suceder un poco antes.

Con el gran auge y despliegue que está alcanzando la banda ancha, los usuarios cada vez consumen más contenidos multimedia en sus ordenadores. Esta tendencia en el consumo de contenidos por banda ancha se ve como una oportunidad por parte de los fabricantes de terminales. Por ello, en 2009, los fabricantes comienzan a desarrollar los primeros *dispositivos híbridos* que posibilitaban el consumo de los contenidos multimedia de internet en el televisor, además de la visualización de contenidos de televisión clásicos.

El acceso a los contenidos de internet en el televisor se llevaba a cabo usando el navegador instalado en el dispositivo. También empiezan a aparecer las primeras aplicaciones propias del fabricante desarrolladas con una variación del lenguaje HTML diseñada para dispositivos de consumo electrónico (*CE-HTML*, 2007).

Las organizaciones impulsoras de los estándares existentes en aquel momento reaccionan a ello. Como hemos hecho en el punto anterior, vamos a repasar las que son plataformas y estándares abiertos.

1.3.1 MHP 1.3 (*GEM 1.3*)

DVB intenta promover la versión del estándar pero fracasó (excepto en Italia). Último estándar ESTI TS 102 727 v1.2.1 (MHP1.3) de Septiembre de 2011: incluye TV híbrida.

1.3.2 MHEG-IC

Otro estándar que intenta evolucionar es el MHEG-5. Para ello, en Marzo de 2009 se publica en el *D-Book* (especificaciones técnicas para la Televisión Digital Terrestre en el Reino Unido) un nuevo componente para MHEG-5 el *MHEG-IC* (MHEG Interaction Channel).

MHEG-IC es una extensión para MHEG-5 desarrollada por el DTG para la distribución de servicios interactivos de televisión por vía IP. Los dispositivos compatibles con MHEG-5 IC se pueden conectar a una red de banda ancha usando cualquier proveedor de servicios de internet para recibir las aplicaciones interactivas y datos asociados.

Además de las capacidades en gráficos y textos de versiones anteriores de MHEG, ahora se amplía con la posibilidad de ofrecer vídeo bajo demanda (*VoD*) y se pueden realizar transacciones seguras para, por ejemplo, ofrecer servicios de comercio electrónico.

En la actualidad está integrado en los sistemas Freeview y Freeview HD.



1.3.3 HbbTV

“UNA NUEVA INICIATIVA EUROPEA FUSIONA LA TV CON EL PODER DE INTERNET.

Un consorcio europeo formado por líderes de la industria de la televisión, anuncia el lanzamiento de Hybrid Broadcast Broadband TV o HbbTV, una iniciativa paneuropea cuyo objetivo es la armonización de la distribución por radiodifusión y banda ancha de noticias, información y entretenimiento para los usuarios finales mediante televisores y set top boxes con una conexión de banda ancha opcional’.

Así se presentaba HbbTV públicamente el 27 de Agosto de 2009 a los medios de comunicación.

HbbTV se presentó como un sistema con el que los usuarios tendrían acceso a nuevos servicios proporcionados por radiodifusores, proveedores de contenidos y fabricantes de dispositivos. Los servicios que se ponían como ejemplo eran el *catch-up TV*, el video bajo demanda, la publicidad interactiva, votaciones, juegos, redes sociales y servicios relacionados con el programa como EPG y teletexto digital. Todos estos servicios ofrecidos mediante cualquiera de las tecnologías de radiodifusión: terrestre, cable y satélite. Las primeras demostraciones de HbbTV tuvieron lugar en la feria *IFA 2008*.

HbbTV proporciona a los radiodifusores una plataforma abierta como alternativa a las tecnologías propietarias que ya estaban presentes en el mercado. El interés por una plataforma abierta se debe a que cada TV híbrida que era introducida en el mercado usaba una implementación ajustada técnicamente al producto que se estaba intentando comercializar: el navegador de la TV soportaba distintos formatos, lenguajes, protocolos... y la filosofía del dispositivo híbrido seguía la estrategia de ofrecer, por un lado, los servicios clásicos de los canales de TV convencionales y, por otro lado, las aplicaciones de internet por el canal de banda ancha. Ofrecidos de esta manera, el usuario debe acceder a los distintos contenidos de forma separada, utilizando un botón específico para cada uno. Se presentan como dos funciones claramente diferenciadas dentro de un mismo dispositivo.

Este es el motivo por el que se inicia la actividad del consorcio HbbTV (inicialmente un pequeño conjunto de empresas), que decide proponer un escenario técnico uniforme que conecte los servicios y funciones de internet con los de la radiodifusión utilizando tecnologías HTML. Los objetivos eran:

- Ser abierto y estar estandarizado para permitir el desarrollo de contenidos, independientemente de la implementación particular de cualquier fabricante.
- Estar basado en el mayor número de tecnologías ya estandarizadas.
- Especificar el conjunto de componentes y funciones mínimo. Así el máximo número de empresas podrían soportar la especificación.
- Hacer funcionar entre sí la mayor cantidad de sistemas de radiodifusión (DVB-T/S/C) y de tecnologías de acceso a internet (ADSL, cable, WiFi).
- Conectar internamente la *TV lineal* y los servicios interactivos adicionales.
- Usar los canales broadcast para la distribución de servicios adicionales.
- Sustituir al teletexto (30 años después) con el llamado teletexto digital.
- No permitir la combinación de servicios de internet con la imagen de TV lineal de forma descontrolada (*overlay*), rompiendo así la integridad editorial.

Por ejemplo, HbbTV permite arrancar páginas HTML desde el programa en emisión, mejorar la calidad visual y opciones del teletexto en la era de la alta definición, incorporar información adicional sobreimpresa (como un “ticker” de noticias), etc. Detengámonos un instante en el último ejemplo, el del ticker. Puede parecer poco novedoso, porque en la práctica totalidad de informativos de TV hay una banda de texto pasando por la parte de abajo de la pantalla con distintas noticias breves ¿qué novedad aporta HbbTV a esto que ya existe? El ticker habitual viene realizado por el radiodifusor en origen, y no es modificable. Ese mismo ticker con tecnología HbbTV podría ser una banda superpuesta al ticker “fijo” al cual el radiodifusor (que es quien también desarrolla ese servicio de ticker HbbTV) le habilita la opción de ser personalizado. Así, el telespectador podría decidir si solo recibir noticias de ámbito local y generales, deportes, viajes y economía y prescindir de cultura, tecnología e internacional. Esto es solo un pequeño ejemplo.

Inicialmente en Mayo de 2009 se publicó la especificación HbbTV 0.5 y luego la 1.0 que fue enviada a la ETSI para su aprobación como estándar. La ETSI la aprobó y publicó como estándar en Junio de 2010, es el estándar ETSI TS 102 796 v1.1.1. A partir de ahí la iniciativa de empresas que impulsa HbbTV se transformó en el consorcio HbbTV.

El 24 de Febrero de 2012 (casi 2 años después) se publica la errata y desambiguaciones de la primera versión del estándar.

El 10 de Abril de 2012 se publica la versión 1.5 del estándar que está en proceso de aprobación por la ETSI y que incorpora streaming adaptativo (*DASH*), *DRMs* y EPG avanzado. La futura aprobación de HbbTV 1.5 por la ETSI dará lugar a la versión 1.2.1 del estándar, que se denominará ETSI TS 102 796 v1.2.1.

El esquema del sistema HbbTV es el siguiente:

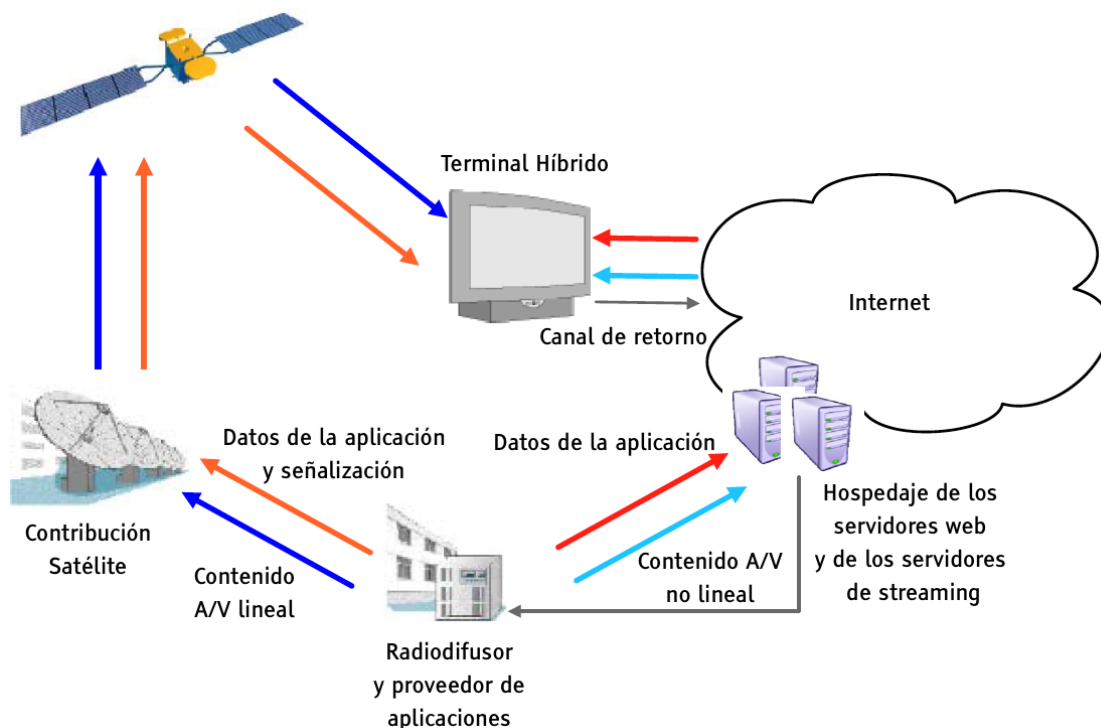


Figura 1.4 – Esquema general de un sistema HbbTV



1.3.4 DTG UK CTV

Como hemos comentado anteriormente, el DTG publica y mantiene el D-Book, que es el libro con las especificaciones técnicas para la televisión digital en el Reino Unido. Los servicios de televisión interactiva en el Reino Unido se ofrecen mediante el estándar MHEG-5 especificado por el D-Book.

Se han invertido muchos millones de fondos públicos en la evolución de los actuales sistemas Freeview y Freeview HD, que son los nombres comerciales de los sistemas de televisión digital terrestre en el Reino Unido. También se ha invertido bastante en el denominado proyecto Canvas y rebautizado como YouView, que pretende sustituir a los anteriores a largo plazo.

Motivado por los avances que se están produciendo en el campo de la televisión híbrida, el DTG ha actualizado sus especificaciones para la televisión conectada. La nueva especificación servirá de base a proveedores de servicios como Youview, Fetch TV, LoveFilm, Sky o Virgin Media para el desarrollo de sus plataformas. La especificación ha sido consensuada por los distintos representantes de la industria.

Para desarrollar la especificación, el DTG ha creado 7 grupos de trabajo divididos en las siguientes áreas: arquitectura, terminal, distribución, presentación, metadatos, seguridad y medidas.

Buscando la armonización con otras tecnologías la UK CTV toma como base la especificación HbbTV a la que hace algunas modificaciones que la hacen algo más potente.

Pero el reto principal que tiene que superar el DTG es que esta nueva tecnología de presentación basada en HTML tiene que coexistir con el inmenso despliegue de aplicaciones y receptores MHEG-5.

El grupo encargado de desarrollar esta tecnología está trabajando en 3 áreas:

- Tecnología de presentación HTML (basada en HbbTV).
- Un perfil de estándares web para mejorar capacidades gráficas.
- Un *framework* que permita la coexistencia de HTML y MHEG-5 con otras tecnologías como Flash.

Como nota adicional a este punto, cabe decir que el 12 de julio de 2012 una organización profesional del sector de la alta tecnología en el Reino Unido llamada Intellect (www.intellectuk.org), ha expuesto a los radiodifusores del Reino Unido su especificación y conjunto de tests para la incorporar la tecnología HbbTV en la plataforma Freeview HD. En breve se van a iniciar las conversaciones con los distintos radiodifusores.



1.4 Web TV, IPTV, Hybrid TV y OTT.

En los inicios de la banda ancha, el video web llegaba casi en exclusiva a las pantallas de ordenador. Podían ser servicios gratuitos de *WebTV* como Youtube o servicios de pago como Hulu o Netflix. Después, ISPs como Telefónica o Ya.com y operadores como Ono comenzaron a usar *IPTV* para ofrecer servicios de televisión utilizando la pila de protocolos *TCP/IP*, la misma que se usa para las comunicaciones en internet. Los servicios llegan al terminal receptor (que es proporcionado por el operador) mediante una red de banda ancha de operador también conocida como *managed network*. La red de operador generalmente utiliza como medio de transmisión una línea telefónica mejorada (ADSL) o una red de cable.

Los fabricantes comenzaron también a fabricar dispositivos híbridos que permitían disfrutar de los contenidos de internet en el TV, enriqueciendo así la experiencia con las TV y proporcionando un tipo de interactividad basada en un portal de aplicaciones. Los radiodifusores (sobre todo públicos) y proveedores de contenidos, haciendo uso de estándares abiertos comienzan a usar los terminales híbridos para ofrecer servicios de teletexto enriquecido, VoD, Catch Up, etc.

Todos estos términos mezclados pueden confundir al usuario porque habitualmente se habla de ellos de forma indistinta. Si bien es cierto que cada uno de ellos tiene una componente de internet y TV, se llevan a término de distinta forma. Vamos a definirlos:

IPTV: Hace referencia a la transmisión de servicios de radiodifusión (habitualmente lineales) mediante una red de operador. Para dar calidad de servicio se emite una señal (de radio y TV) encapsulada en protocolo IP en un espacio reservado de la conexión de banda ancha del operador, generalmente una ADSL o cable. Para poder reproducir el contenido, hace falta un terminal proporcionado por el operador capaz de decodificar la señal y presentarla en el televisor además de una suscripción/contrato.

WebTV: Este término hace referencia a todos los contenidos multimedia distribuidos a través de la web sin calidad de servicio. Estos contenidos suelen ser presentados en ordenadores, tabletas, smartphones o reproductores multimedia conectados a la TV. Ya se han comercializado TV con SO Android (Google TV) con/sin sintonizador que hacen estas funciones.

Terminales Híbridos: Son dispositivos capaces de mostrar las señales que les llegan por canales broadcast DVB-T/C/S junto con contenidos que llegan por la conexión de banda ancha (algunos de estos contenidos son del tipo WebTV). Este tipo de dispositivos además suelen tener capacidades de conexión y distribución de contenidos en el entorno doméstico (DLNA) y el propio fabricante ofrece ya aplicaciones para controlar el terminal desde teléfonos móviles. Por ejemplo, una SmartTV es un terminal híbrido.

OTT: *OTT* (Over The Top), es un término cuyo uso se ha extendido mucho en los últimos tiempos, debido a la convergencia de internet, la televisión y los contenidos multimedia. El término se usa para hablar de contenidos multimedia (generalmente de vídeo) cuya distribución se hace utilizando las infraestructuras de un operador de internet (ISP). Por ejemplo, el VoD de una aplicación HbbTV o el video de la aplicación de Youtube de un portal de fabricante de una SmartTV se hacen "Over The Top".

Podemos intercambiar los términos Televisión Conectada (más comercial) y Televisión Híbrida (más técnico). Cuando usamos cualquiera de los términos, en el fondo estamos hablando de un terminal conectado a internet que, además de los contenidos de TV lineal, son capaces de ofrecer una experiencia interactiva que puede estar centrada en las aplicaciones del portal, en los contenidos ofrecidos por los radiodifusores o en ambas.

En el siguiente gráfico intentamos mostrarlo todo:

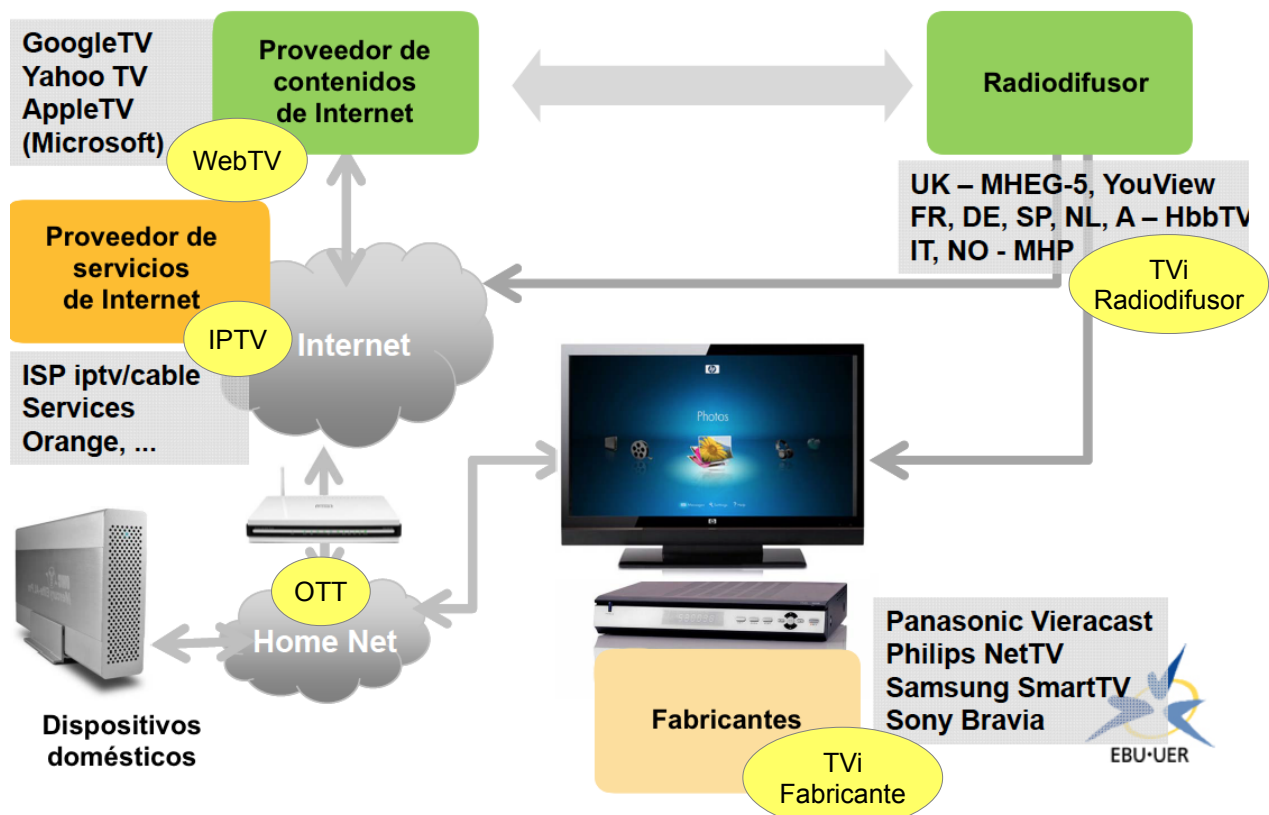


Figura 1.5 – Representación de las distintas tecnologías distribución web, IP, híbridas e interactivas

Como curiosidad, una forma visual de diferenciar las aplicaciones interactivas del portal del fabricante de las aplicaciones interactivas ofrecidas por el radiodifusor: según el botón del mando a distancia que se use para acceder a cada una.

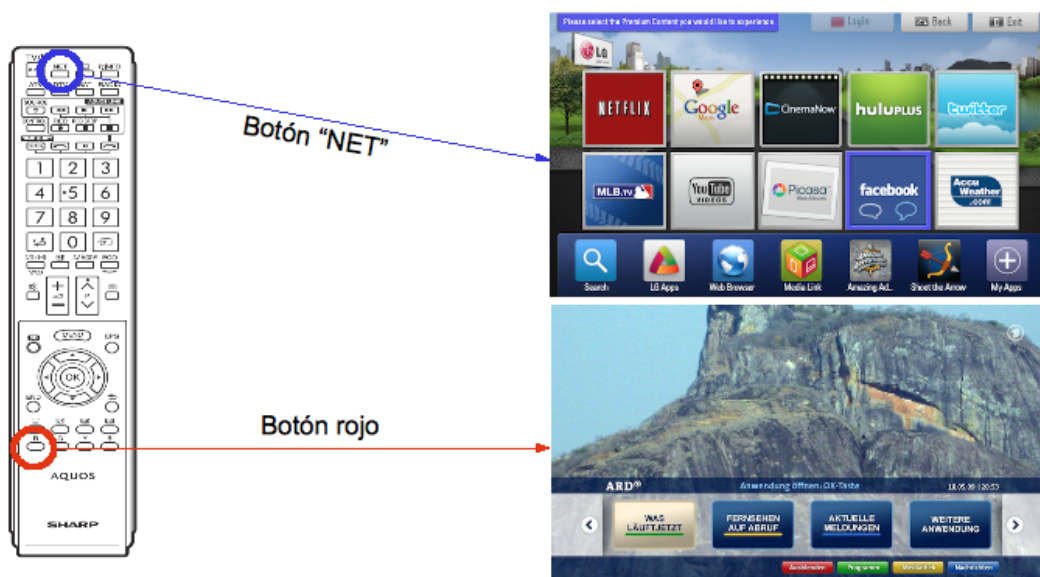


Figura 1.6 – Diferencia entre una aplicación HbbTV y una aplicación SmartTV de fabricante

1.5 Comparativa de estándares

	MHEG	MHP	HbbTV	Propietarios
¿Está estandarizado?	SI	SI	SI	NO
¿Es abierto?	SI	SI	SI	NO
¿Derechos por uso?	NO	SI *	Los reclaman, pero no van a influir en el despliegue del estándar como ocurrió con MHP.	Acuerdos comerciales con el fabricante para incluir tu aplicación en su portal.
Nombre de la especificación	ETSI ES 202 184 v2.1.1 (perfil MHEG-5 UK v1.06)	ETSI TS 102 727 v1.2.1 (MHP1.3)	ETSI TS 102 798 v.1.1.1 En camino la v.1.2.1 (erratas) y tras ella la 1.5	Algunos fabricantes se han unido en un grupo llamado SmartTV Alliance que promulga un SDK común para el desarrollo de las aplicaciones.
¿Permite TV interactiva?	SI, basada en aplicaciones de radiodifusor.	SI, basada en aplicaciones de radiodifusor.	SI, basada en aplicaciones de radiodifusor.	SI, basada en aplicaciones de portal.
¿Es híbrido?	En la última versión	En la última versión	Desde su origen	Se ejecuta en un terminal híbrido
Ventajas	<ul style="list-style-type: none"> - Creado desde cero para ser interactivo - Ampliamente desplegado y probado en UK - Ha evolucionado con el tiempo 	<ul style="list-style-type: none"> - Simple de implementar - Usa Java, un lenguaje bien conocido - Ha evolucionado con el tiempo 	<ul style="list-style-type: none"> - Nace ya híbrido - Usa tecnologías ampliamente conocidas y probadas - Totalmente enfocado al radiodifusor y a sus intereses 	Como experiencia de internet con el TV las tiene todas. Son portales muy cuidados que acercan eficientemente al usuario los servicios de internet mediante la TV.
Inconvenientes	<ul style="list-style-type: none"> - Es un lenguaje de desarrollo totalmente independiente de cualquier otra tecnología ya existente - Necesita de desarrollos específicos siempre, no hay una forma rápida de portar aplicaciones hechas con otras tecnologías ya existentes (HTML) aunque soporta este lenguaje 	<ul style="list-style-type: none"> - Está gravado con derechos de propiedad intelectual - Por ello cayó en desgracia entre todos los países que habían apostado por él (con la excepción de Italia) - Es como el latín, un lenguaje "muerto" 	<ul style="list-style-type: none"> - Realmente está extendido solo en Alemania y en Francia. España ya empieza a lanzar cosas y hay países que muestran interés pero su situación en el mercado es algo vulnerable. Debe consolidarse lo antes posible. 2012 debe ser el año. 	Para los desarrolladores de aplicaciones significa 1 fabricante → 1 desarrollo específico (pendientes de Smart TV Alliance) Posicionamiento de marca dentro del portal (¿pondrán mi aplicación en la 1ª página o en la 5ª?)
¿Donde se usa?	Reino Unido, Nueva Zelanda, Australia, Hong Kong,	Italia	Alemania, Francia, España, Dinamarca, Austria.	Global, en cada TV conectada.

- http://www.dvb.org/news_events/news/archive/via_licensing_releases_fi/index.xml



2. HbbTV

En este apartado vamos a centrarnos exclusivamente en el estándar HbbTV, ya hemos adelantado algo en el punto anterior de la introducción, pero en este apartado se hace un análisis más detallado de todas las motivaciones, implicaciones e intereses que están detrás del estándar. También entraremos en cuestiones más técnicas pero no de forma muy profunda, algo que dejamos para el siguiente bloque.

2.1 ¿Por qué se crea este nuevo estándar?

HbbTV es una iniciativa paneuropea que busca ofrecer contenidos broadcast y broadband/web en la pantalla del televisor. En la práctica, la especificación HbbTV solo aporta un par de componentes técnicos novedosos, el resto está basado en estándares ya existentes. Lo que hace es detallar un perfil sobre la base de las tecnologías que lo componen, no desarrolla desde cero una tecnología para aplicarla a la TV. Habiendo elegido este camino, para la creación del estándar HbbTV es posible conseguir dos hitos muy importantes: (1) abaratar al máximo los costes de desarrollo y (2) reducir al máximo el “*time-to-market*”, es decir, el tiempo necesario para colocarlo en el mercado.

Los fabricantes en torno al año 2009 comenzaron a fabricar terminales híbridos que permitían seguir disfrutando de los contenidos lineales ofrecidos por un canal DVB-T/C/S y a esto se le añadía la posibilidad de disfrutar en el televisor de contenidos de internet.

Todos los fabricantes tienen hoy al menos un terminal híbrido en catálogo. Es un nicho de mercado en el que cualquier fabricante considera que debe estar representado. Lo que suelen incorporar en esos terminales híbridos son pequeñas aplicaciones como Youtube, periódicos online y contenidos Catch Up de los principales radiodifusores... contenidos que el radiodifusor pone a disposición del fabricante realizando una pequeña aplicación que debe ser compatible con su plataforma y que luego el fabricante coloca donde considera más oportuno, acuerdos comerciales aparte. El radiodifusor, para tener visibilidad en el mercado, debería hacer una aplicación por cada uno de los dispositivos que cada fabricante tiene en el mercado, al menos para los que más cuota de ventas tengan. Cuando el fabricante cambia el portal y con él algunas características internas de funcionamiento hay que actualizar/rehacer la aplicación.

Con esta filosofía se altera el orden jerárquico entre el fabricante y el radiodifusor. Siempre ha sido el fabricante el que ha trabajado por ofrecer de la mejor manera posible los contenidos que el radiodifusor ha puesto en servicio, intentando conseguir el mejor dispositivo, el que ofreciese el mayor número de canales con todos sus servicios y con la mejor calidad.

Con la filosofía del portal y plataforma propietaria, es el fabricante el que le abre la puerta al radiodifusor ofreciéndole la posibilidad de aparecer en su plataforma, pero acordando las condiciones en las que esto sucede.

Según este esquema, el radiodifusor se enfrenta al problema de que para poder estar en todas las plataformas (de operadores y fabricantes) debe someterse a una serie de exigencias técnicas que en la práctica es muy difícil de poder cumplir.

Los tres primeros problemas que podemos identificar son:

- 1) Cada fabricante utiliza su propio navegador/plataforma con un perfil adaptado a los servicios que pretende comercializar y con el conjunto de características técnicas que haya querido implementar como, por ejemplo, protocolos de streaming, resoluciones, tasas binarias, codificaciones o conjuntos de teclas del mando a distancia. Como ya hemos comentado antes, esto significa un desarrollo por fabricante que a veces no se limitaba a reescribir la aplicación para otro navegador, también pasaba que había que adaptarse al concepto de TV híbrida del fabricante (mandos con más o menos botones, navegación entre menús no homogénea...) Al proveedor de contenidos esta fragmentación técnica le resulta un obstáculo.
- 2) El portal de entrada a las aplicaciones de internet del fabricante en las que se mezclan aplicaciones de radiodifusores, periódicos online, páginas web, redes sociales, etc. está casi totalmente controlado por el fabricante del televisor. Los consumidores no pueden añadir o quitar contenidos con total libertad a ese portal.
- 3) En la estrategia de televisión híbrida del fabricante lo único que tenían en común el canal broadcast y el broadband era el monitor de TV. Por lo demás son facetas completamente independientes de un mismo dispositivo. Cuando se visualiza un programa de televisión, se pierde la navegación y cuando está activa la parte de internet se perdía la televisión en directo o la imagen de canal de televisión se sobreimpresionaba con contenido que no tenía nada que ver con la emisión. Un ejemplo:

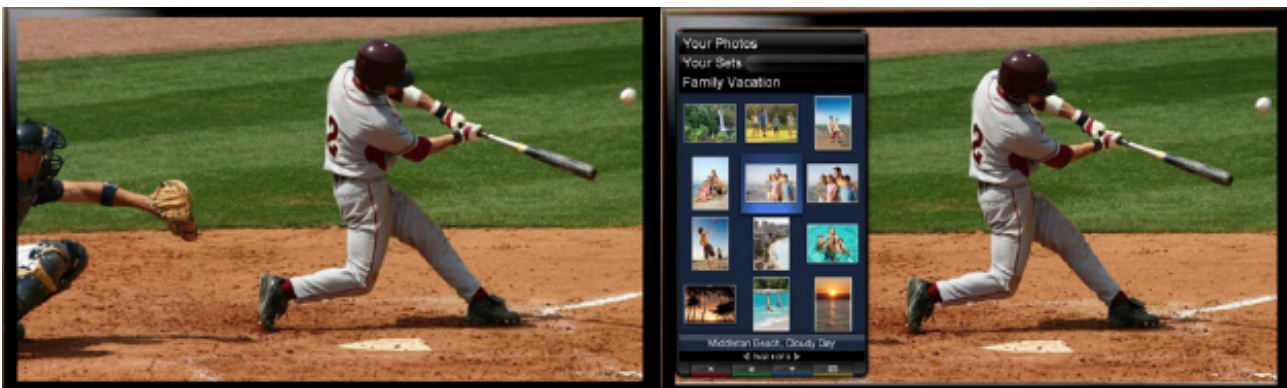


Figura 2.1 – Ejemplo de una sobreimpresión (overlay) sobre un contenido broadcast.

Por ello, en Febrero de 2009 la *EBU* (European Broadcasting Union) en su documento “EBU Recommendation R127” anuncia:

“EBU recomienda a sus miembros que fomenten, en cooperación con la industria y los organismos de estandarización, el desarrollo de plataformas tecnológicas híbridas con la mayor cantidad de componentes técnicos comunes, para garantizar el desarrollo de un mercado de consumo a nivel europeo, con el mínimo coste y la más alta calidad para los consumidores. Con el fin de lograr este objetivo, la EBU establecerá un Grupo de Trabajo con la representación más relevante de la industria.”

En enero de 2010 la EBU elaboró un documento denominado “Requirements for the standardization of Hybrid Broadcast/Broadband (HBB) Television Systems and services.” (<http://tech.ebu.ch/docs/tech/tech3338.pdf>).



Es un documento genérico acerca de los requisitos que se consideran básicos en los sistemas híbridos como tal, no hace referencia ni es derivado del estándar HbbTV (que aún no era estándar). Casi un año después la WBU (World Broadcasting Unions) formada por todos los grupos de radiodifusores a nivel mundial analizó el repunte del fenómeno híbrido y sacó un documento que reflejaba su posición en base a las discusiones llevadas a cabo (<http://www.nabonet.com/wbuarea/committees/tc.asp>).

Basado en este documento, la EBU elaboró otro para Europa denominado “Principles for internet connected and Hybrid Television in Europe”, en el que aparece un concepto interesante: “televisión híbrida” (no “terminal híbrido”), este es el término que mejor refleja el punto de vista del radiodifusor, es la forma de decir que sus contenidos televisivos llegan por más vías además de la tradicional. HbbTV se ajusta perfectamente a este término. Los principios que la EBU enumera en este documento son:

Para que el sistema no fracase de inicio y sea un éxito potencial:

- 1 – Piden que las plataformas y dispositivos se adapten a los sistemas, así los radiodifusores pueden establecer el vínculo entre el contenido lineal y el no lineal.
- 2 – Hay que facilitar al espectador el acceso a los contenidos no lineales mientras ve los canales lineales, y deben ser fáciles de encontrar y navegar por ellos.

En cuanto a la integridad de los contenidos y la representación de la señal en pantalla:

- 3 – Los *servicios lineales* y no lineales del radiodifusor deben presentarse en los dispositivos híbridos sin ninguna alteración ni ninguna interrupción que altere la experiencia audiovisual. La señal (video y audio) no debe ser alterada sin que el usuario con una acción voluntaria tome esta decisión.
- 4 – Los sistemas híbridos no deben permitir la superposición (*overlay*) de contenido de terceros o de contenidos publicitarios sobre la señal de televisión sin el consentimiento del radiodifusor o como respuesta a una acción activa por parte del usuario.
- 5 – Se aplicará el punto anterior a cualquier tipo de comunicación comercial que aparezca en cualquier posición de la pantalla a la par que la señal de televisión.

Los puntos 3, 4 y 5 son los que causan mayor conflicto entre fabricantes y radiodifusores a nivel europeo.

En cuanto al acceso al contenido del radiodifusor:

- 6 – Los portales/menús de las televisiones híbridas (propietarios) deben garantizar un acceso no discriminatorio a cualquier radiodifusor o proveedor de contenidos.
- 7 – La oferta no lineal del radiodifusor debe ser claramente representada en pantalla y fácilmente accesible y ubicada en una categoría apropiada.
- 8 – Los usuarios deben poder personalizar listas de canales favoritos
- 9 – Los usuarios deben poder acceder a las aplicaciones de portal de los radiodifusores desde la pantalla principal. La aplicación estará identificada con el logo del



radiodifusor. También podrán acceder pulsando el “botón rojo” mientras ven uno de sus canales de televisión.

10 – Si el sistema híbrido incluye un motor de búsqueda, los contenidos lineales/no lineales del radiodifusor deberán referenciarse correctamente utilizando los metadatos proporcionados por el radiodifusor.

11 – El usuario debe poder volver al último canal visto mediante un botón “retorno” específico.

En cuanto a la preservación del entorno de visualización, incluyendo la protección de menores:

12 – Los sistemas híbridos no pueden ser usados para no cumplir la regulación de las actividades de radiodifusión. Se deben respetar las leyes que tienen como objetivo la protección del menor, y deben facilitarse los controles parentales.

13 – La fuente de los contenidos debe estar claramente identificada sin perjuicio de que el usuario pueda añadir contenidos procedentes de otras fuentes.

Derechos de propiedad intelectual y piratería:

14 – Los radiodifusores deben poder pedir que se retiren las aplicaciones que faciliten el acceso a contenido sin consentimiento o derechos de sus autores.

15 – El radiodifusor deberá tener la opción de usar cualquier sistema de DRM para proteger su contenido lineal/no lineal sin la necesidad de codificar la señal.

16 – Los logos e identificativos del radiodifusor deberán estar siempre visibles durante la visualización del canal, esto no prohíbe que el usuario cambie la apariencia de visualización en pantalla si así lo desea.

Protección de datos de usuario:

17 – Deben cumplir fielmente con las leyes nacionales, europeas e internacionales en materia de protección de datos.

18 – Los datos personales que contengan algo más que información operacional (p.ej para realizar pagos) solo se recogerán tras un consentimiento explícito por parte del usuario. La creación de perfiles de usuario enlazadas con IPs (incluida la geolocalización) exigirá que el usuario muestre su consentimiento expreso sobre un aviso legal.

19 – Los radiodifusores podrán ser informados y, bajo petición, tener acceso a los datos que el sistema híbrido pueda haber recogido en relación al uso de sus servicios, siempre y cuando se respeten las leyes de protección de datos.



En estos principios se puede observar como se mezclan consideraciones acerca de la televisión híbrida junto con consideraciones relacionadas con los portales y contenidos de fabricantes por canal broadband. El documento de la EBU no es restrictivo y ha abordado este tema dejando una puerta abierta al fabricante para que pueda explotar su negocio intentando que eso no afecte a los contenidos del radiodifusor. Tampoco hace referencia a ningún estándar en concreto como HbbTV (que aun no era estándar), aunque posiblemente si que estuviese influenciado por las nuevas iniciativas europeas.

A principios de 2009, casi en paralelo al primer documento de la EBU (la recomendación R127) un conjunto de representantes de varios sectores interesados, se unieron a un proyecto paneuropeo (HbbTV) que estaba desarrollando un sistema que pretendía solucionar los problemas que habían identificado los radiodifusores para el desarrollo de la TV híbrida. Los objetivos de este sistema fueron:

- Que estuviera basado en tecnologías HTML. Así se puede ser eficiente al desarrollar contenidos reutilizando tecnologías ya existentes sin estar ligado a ningún fabricante, operador, etc...
- Usar componentes estandarizados en la mayor medida de lo posible, buscando que el sistema sea fácilmente aceptado y que el tiempo de puesta en el mercado (time-to-market) sea el menor posible.
- Especificar un conjunto de características mínimo para los requisitos básicos. Esto permite una fácil integración con las plataformas hardware ya existentes y evita el rechazo.
- Permitir la combinación de todos los tipos de redes de radiodifusión y de todas las tecnologías de acceso a internet. DVB-T/C/S, WiFi, LAN, PLC...
- Que el sistema híbrido usara recursos del canal broadcast y del canal broadband al mismo tiempo.
- Que permitiera modernizar y sustituir el actual sistema de teletexto. Quizá parezca que este objetivo es un poco “obvio” o que se le da demasiada importancia al teletexto pero tiene una explicación. En 2009, que es cuando se sucede todo esto, hay algunos países pendientes de hacer la transición de analógico a digital y que no usaban ningún estándar de TVi. Para que estos países se decantasen por HbbTV en el momento de elegir un estándar para aplicar a su TVi, era necesario asegurar que iban a tener una alternativa a sus sistemas de teletexto analógico.
- Proteger de la piratería las señales de TV por parte de páginas web.
- Que se pudiese aplicar también a la radio.

Un sistema como HbbTV permite aplicaciones interactivas que estén ligadas a un programa de TV. Aunque HbbTV no es el primer estándar de televisión interactiva, si que es el primero que nace como estándar puramente híbrido. ¿Tendrá éxito o por el contrario caerá en desuso como ya ha ocurrido en el pasado con estándares anteriores? Puntos a favor:

- HbbTV pretende generar una situación ventajosa aprovechando la tecnología ya existente.
- Define los requisitos mínimos. No obliga al fabricante a hacer implementaciones caras, complejas y que requieran de desarrollos que solo sirven para este entorno.
- Crear una solución que sea extensible para permitir la diferenciación desde el punto de vista del fabricante. El estándar HbbTV contempla bastantes características adicionales que son opcionales (*PVR*, descargas, *TimeShift* ...)
- El desarrollo de aplicaciones es rápido porque se pueden probar usando un PC.
- No es pesado de ejecutar en el dispositivo (al menos no más que las otras aplicaciones preinstaladas por el fabricante).



- No depende de un portal centralizado y la señal de radiodifusión es desde donde se arranca la aplicación, algo que mantiene al programa de TV como elemento central del sistema.

HbbTV a priori reúne las condiciones necesarias para que sea ampliamente adoptado o al menos seriamente considerado.

Como mencionamos anteriormente, el consorcio HbbTV remite a finales de 2009 las especificaciones de este sistema a la ETSI que lo publica como estándar en junio de 2010 → ETSI TS 102 796 v1.1.1.

El consorcio en sus orígenes (antes de ser consorcio) nació como un acuerdo entre una iniciativa alemana y una iniciativa francesa. Los primeros miembros fueron ANT, APS, Canal +, EBU, la televisión pública francesa, IRT, OpenTV, Philips, Sony y Samsung.

Se estableció como consorcio en 2010 justo tras aprobarse el estándar europeo y en la actualidad lo forman unos 60 miembros.

Está estructurado en los siguientes subgrupos:

- Grupo de dirección: Gestiona el consorcio.
- Grupo técnico y de especificaciones: 3 grupos de trabajo: errata HbbTV 1.1.1 (terminado), HbbTV 1.5 (terminado) y HbbTV 2.0 (en progreso).
- Grupo de Tests: Desarrollan baterías de pruebas para la especificación.
- Grupo de certificación: Establecen los procedimientos para la obtención del certificado y el uso del logo.
- Grupo de marketing: preparan material para el desarrollo del *B2B*.

Como resultado del trabajo desarrollado por el consorcio HbbTV ya han sido publicados:

1 – HbbTV versión 1.0 (Estándar ETSI TS 102 796 v1.1.1)

2 – Errata del HbbTV 1.0 (corrección de errores, desambiguaciones, no hay nada nuevo).

3 – HbbTV 1.5. Esta especificación es una reacción a los requerimientos del mercado a corto plazo (DASH y DRMs principalmente). Añade algo al HbbTV 1.0. Aún no ha sido estandarizado pero se está a la espera de que la ETSI lo publique como el estándar ETSI TS 102 796 v1.2.1.

2.2 Canal Broadcast y canal Broadband.

Como ya hemos comentado, el estándar HbbTV permite que un terminal híbrido esté conectado en paralelo a dos redes distintas:

- **Broadcast:** (por ejemplo DVB-T, DVB-S, DVB-C): A través de esta conexión a la red broadcast el terminal híbrido recibe la señal de TV convencional (también llamada contenido A/V lineal o TV lineal) junto con la señalización de aplicaciones y los datos de la aplicación. Aunque el terminal no esté conectado a la otra red (la red broadband, red de banda ancha) esta conexión a la red broadcast permite que el terminal reciba aplicaciones ligadas al broadcast (broadcast-related). Además la señalización de los llamados “Stream events” permite la sincronización de aplicaciones broadcast-related con contenido A/V lineal. Esto es útil para lanzar una encuesta o pregunta en un momento determinado, un anuncio, etc... pero en este caso sin un canal de retorno (en el caso del HbbTV el de banda ancha) la interactividad queda limitada a una interactividad local.
- **Broadband (banda ancha IP):** Además, el terminal puede estar conectado a internet a través de un interfaz de red de banda ancha. Sorprende un poco que esto sea algo opcional en el estándar. Así se posibilita la comunicación bidireccional con el proveedor de aplicaciones (que puede ser directamente el radiodifusor o un tercero). En este interfaz, el terminal puede recibir los datos que componen la aplicación así como contenido A/V no lineal (es decir, *streaming* y video On Demand). Además como característica opcional, el terminal puede soportar la descarga de contenidos.

El siguiente esquema muestra un sistema compuesto por un terminal híbrido (la TV), con una conexión broadcast DVB-T y una conexión de banda ancha (ADSL, Cable, FTTH...):

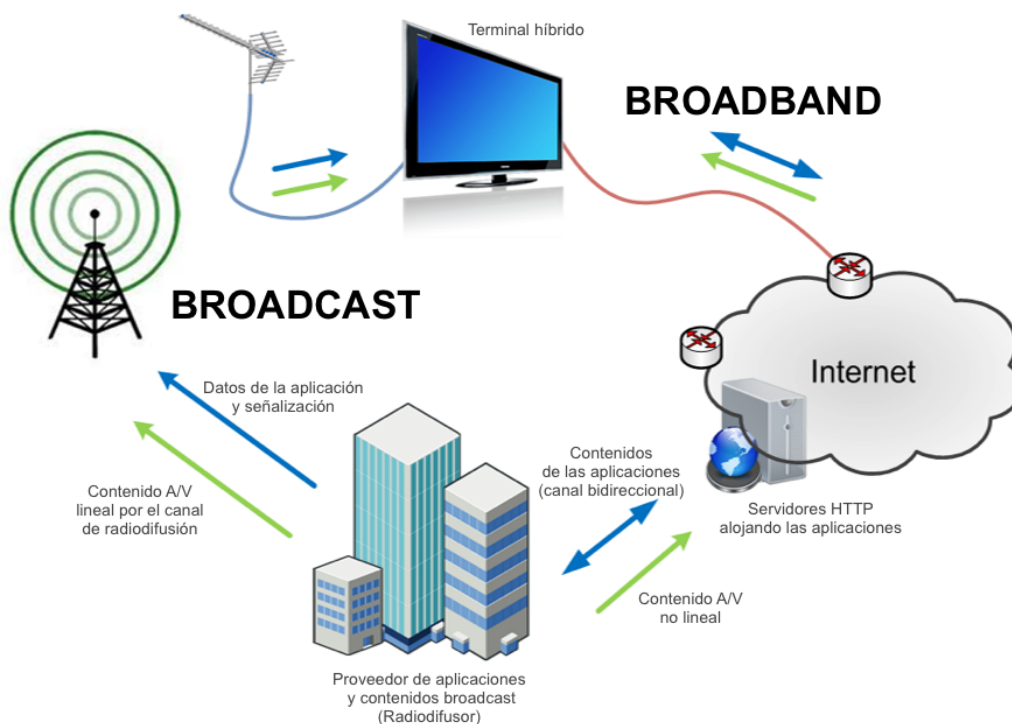


Figura 2.2 – Esquema general de un sistema HbbTV.



2.3 Tipos de servicios que posibilita HbbTV

HbbTV es una tecnología que aglutina muchas de las tecnologías ya usadas en la Web, algo que hace posible ofrecer una gran variedad de servicios debido a la experiencia que ya se tiene con ellas (conocido como *know-how*). Los tipos de servicios que ofrece HbbTV se pueden agrupar en dos categorías: servicios broadcast-related (bc-related, ligados/conectados a la emisión) y broadcast-independent (bc-independent, servicios que pueden ser lanzados desde un servicio bc-related pero que no tienen porqué tener una relación directa con la emisión o el programa).

2.3.1 BC-related

Aquí se engloban las aplicaciones que van señalizadas en la señal de TV según el estándar ETSI TS 102 809 (V1.1.1). La descarga de los datos de las aplicaciones puede hacerse desde el TS de la señal de TV (total o parcialmente) o por el canal broadband.

Ejemplos de aplicaciones BC-related:

- Aplicación de botón rojo como:
 - Resultados deportivos
 - Votaciones, apuestas, encuestas
 - Publicidad interactiva
- Servicios relacionados con el programa visualizado en ese momento (serie, concurso, retransmisión) por ejemplo el T-commerce. (TV commerce, venta de productos a través de la televisión con canal de retorno digital, en este caso internet).
- Teletexto digital que puede ser lanzado desde un botón específico del mando a distancia o accedido indirectamente a través de una aplicación.
- Información de programa: El radiodifusor controla la apariencia y presentación de la información en la pantalla. Esto actualmente se hace delegando en la TV, que lee del transport stream (TS) la información incluida por el radiodifusor y la presenta según el diseño de la aplicación que haya desarrollado el fabricante.

2.3.2 BC-independent

Son aplicaciones cuya carga y lectura de datos solo puede hacerse a través del canal broadband y que no pueden ser señalizadas en el TS. Pueden ser proporcionadas por el radiodifusor, por terceros, por fabricantes de dispositivos, etc.

Ejemplos de aplicaciones bc-independent:

- Juegos
- Redes sociales, fotos compartidas...

Esta jerarquía de aplicaciones y el hecho de que las aplicaciones bc-related necesariamente tienen que ir declaradas en la señalización DVB asegura al radiodifusor que las únicas aplicaciones que irán asociadas a su canal serán las suyas. Por otra parte, las aplicaciones bc-independent permiten una gran variedad de servicios que se pueden añadir a las bc-related. Vamos a ver de forma gráfica estas dos categorías de aplicaciones:

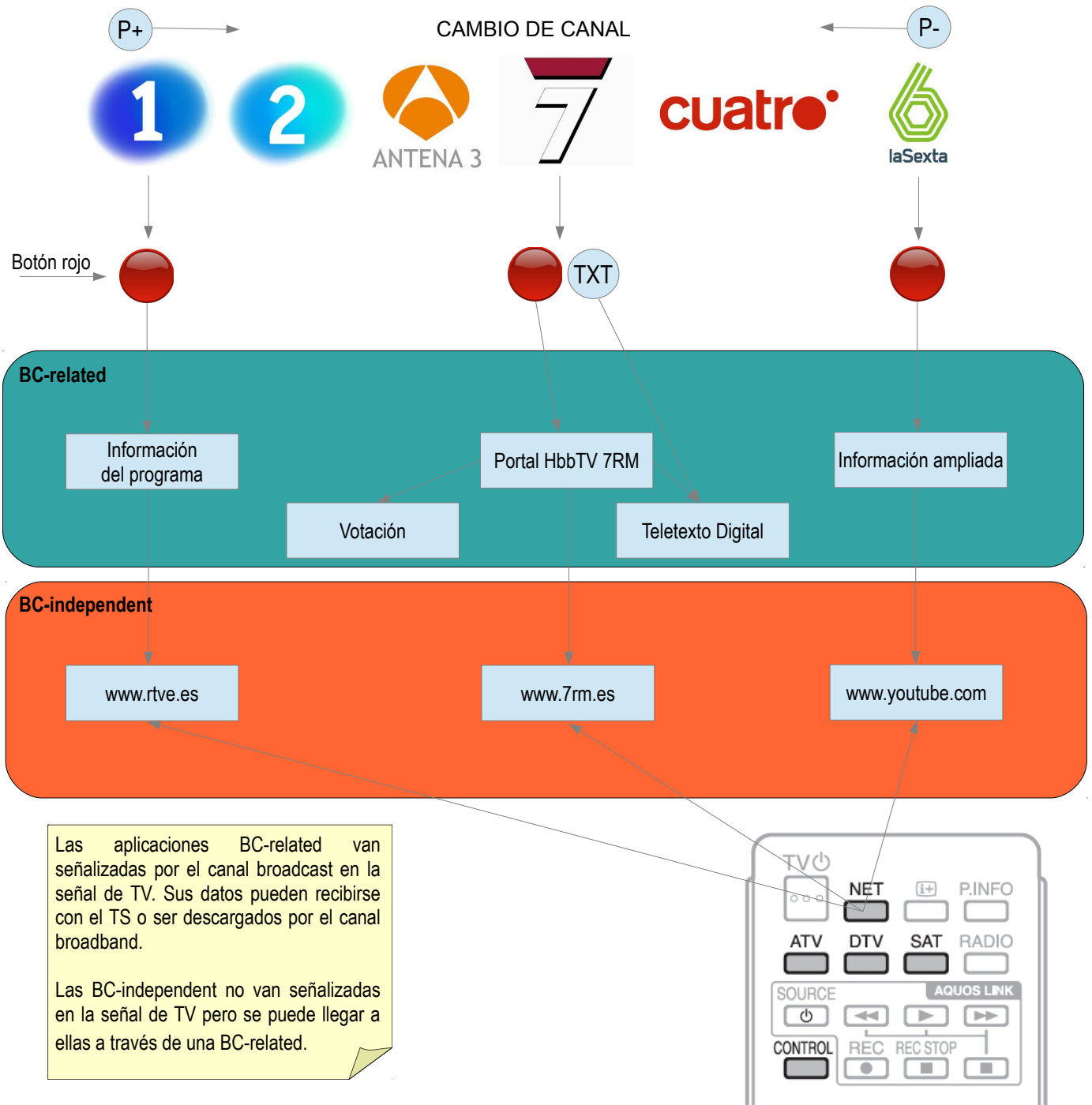


Figura 2.3 – Representación gráfica de las aplicaciones bc-related y bc-independent.



Las aplicaciones como ya hemos mencionado anteriormente pueden ser descargadas por el TS del canal broadcast y por el canal broadband:

- Si se descargan por el canal broadcast dentro del TS, lo hacen mediante un protocolo llamado *DSM-CC*. También se suele hacer referencia a la descarga por canal broadcast como descarga por “protocolo DVB”.
- Si lo hacen por el canal broadband lo hacen usando la pila de protocolos TCP/IP de internet. O también “protocolo IP”.

Para ayudar a comprender los dos tipos de aplicaciones es importante tener en cuenta por una lado la señalización de la aplicación y por el otro por donde se descargan los datos y por donde la aplicación accede a ellos.

Vamos a intentar clarificarlo un poco apoyándonos en la siguiente tabla, en la que introducimos conceptos que veremos más adelante en el punto 4 “Desarrollo técnico del estándar”.

	Señalización	Descarga aplicación		Acceso datos		Tipo de arranque	Ciclo de vida
		IP	DVB	IP	DVB		
bc-related	AIT	SI	SI	SI	SI	AUTOSTART MANUAL	Al seleccionar el servicio Mientras está el servicio seleccionado
bc-independent	Ninguna o AIT XML	SI	NO	SI	NO	Botón portal <i>application.createApp()</i>	Según se haya creado Según lo que haga

Tabla 2.1 – Propiedades de las aplicaciones bc-related y bc-independent.

2.3.3 Ejemplos de servicios



Figura 2.4 – Ejemplo de una aplicación de teletexto mejorado (también llamado teletexto digital).

Esta es la típica aplicación bc-related con el teletexto digital del radiodifusor. Se puede llegar por un enlace en el portal al que se accede con el botón rojo o directamente pulsando el botón “TXT” del mando a distancia.



Figura 2.5 – Ejemplo de una aplicación de servicio meteorológico.

Esto podría ser una bc-independent a la que se llega por un enlace en el portal bc-related del radiodifusor.



Figura 2.6 – Ejemplo de anuncio interactivo. Un botón rojo durante el anuncio nos lleva a este microportal.

Esto es un ejemplo de aplicación comercial. Durante el anuncio convencional del coche, aparece un “botón rojo” ofreciendo más información. Si lo pulsas te lleva a este mini portal con más información, vídeos y solicitud de prueba del coche. Todo desde el sillón.

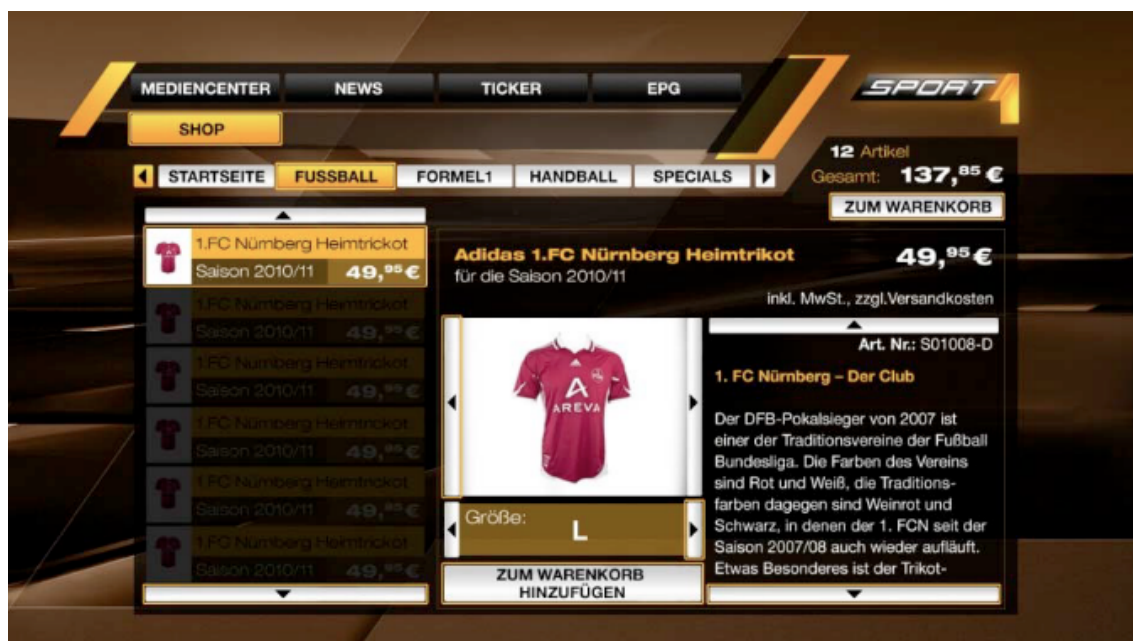


Figura 2.7 – Ejemplo de una aplicación con tienda.

Este canal de deportes tiene un servicio de T-commerce donde vende distintos productos relacionados con el deporte (fútbol, Fórmula1, Balonmano, ...)

Dreiste Musik aus dem Fritz Studio: 02.06.2012 | 20:36

Zeit	Artist(s), Titel
19:57	Troupe At Least We Have Friends (tb)
19:42	Musikredaktion Song
19:39	Foals Spanish Sahara
19:35	Maroon 5 Payphone

Fritz rbb
Und das hört man
ARD Start: 0-Taste drücken

Figura 2.8 – Ejemplo de una aplicación que adelanta la lista de reproducción de una emisora de radio.

Este sería un ejemplo de portal bc-related de una emisora de radio donde se ve una pantalla con capturas de fotos del estudio y se da información de la escaleta y los metadatos de la canción que está sonando (Artista, título, ...)

K i kinderkino.de

Aktuelles Serien **Filme** HD Filme Kostenlos Anmelden Exit

Grid of movie posters including: Asterix und die Nibelungen, Das Buch der Dschunzeln, Die kleine Raupe Nimmersatt, Die Mummies, Lotte, Mullewapp, Kater Zorbas kleiner Mäwe, Winnie der Pooh, and others.

Figura 2.9 – Ejemplo aplicación para la venta de contenidos on-demand.

Este es el portal de VoD de un canal de pago. Ofrece películas y series para consumo on demand. Al seleccionar cualquiera de las películas solicita usuario y contraseña.



Y ahora dos ejemplos más. Son dos recortes de noticias de la revista Panorama Audiovisual:

<http://www.panoramaaudiovisual.com/>

Primera experiencia de publicidad interactiva bajo HbbTV en TF1

Durante la emisión del spot interactivo de una compañía de seguros, los espectadores franceses con televisores conectados podrán con un solo clic acceder a un mini-site de una compañía de seguros.

Por primera vez, gracias a una solución de Wiztivi basada en el estándar HbbTV, TF1 pondrá en marcha a partir del 3 de junio una campaña de publicidad experimental interactiva en tiempo real.

Durante la emisión del spot interactivo de una compañía de seguros, los espectadores con televisores compatibles con esta norma podrán con un solo clic acceder a un site en el que el anunciante ofrece su seguro de automóvil "pay as you drive". Además, si lo desea, el usuario puede contactar directamente con el centro de atención telefónica de la aseguradora o dejar sus datos de contacto para recibir una oferta promocional.



Además de los televisores híbridos, también podrán interactuar con esta campaña los smartphones mediante el escaneo de un código QR para ir directamente al mini-site.

En televisión, el menú desplegable aparecerá superpuesto y sincronizado con la señal en vivo, siendo sólo visible para los espectadores con un televisor conectado HbbTV.

Primer Roland Garros en HbbTv

France Télévisions, con IBM y Panasonic como socios tecnológicos, enriquecerá la emisión del torneo Roland Garros en un entorno híbrido HbbTv.

Figura 2.10 - Publicidad interactiva HbbTV.



La televisión híbrida basada en HbbTV dará un nuevo paso cuando el 29 de mayo arranque el popular torneo Roland Garros. France Télévision, con IBM y Panasonic como socios tecnológicos, desarrollará una ambiciosa cobertura apoyándose en la televisión híbrida y el estándar HbbTv.

Este planteamiento híbrido permitirá a los espectadores no sólo seguir la emisión en directo sino también acceder a una gran cantidad de información y contenidos exclusivos como estadísticas en tiempo real, información de última hora e imágenes de los participantes.

France Television cubrirá un total de 130 horas a través de sus canales France 2, France 3 y France 4, incluyendo emisiones en estereoscópico gracias a los sistemas de captación 3D de Panasonic.

Esta primera experiencia con HbbTV en Roland Garros se ha puesto en marcha en apenas tres semanas. Para acceder a esta experiencia enriquecida que ofrece la televisión híbrida, el espectador únicamente deberá pulsar el botón de acceso de su televisor inteligente en aquellos programas identificados con el logo de HbbTV.

Junto al contenido de media, France Television ha desarrollado una aplicación sobre HTML 4+JavaScript, ya que se ha optado por no emplear HTML 5 al no estar aún estandarizado.

En el desarrollo ha intervenido WizTIVI que ha diseñado la interfaz de usuario y recoge datos a través de servidores de IBM que gestionan la información deportiva de la competición, y entonces le inyecta los datos en el stream a través de Internet (plataforma en la nube escalable de Wiztivi en Amazon), así como la señal de emisión.

Figura 2.11 - Aplicación HbbTV Roland Garros.



2.4 Tecnologías implicadas

El estándar HbbTV está basado en tecnologías y estándares bien conocidos. Usando cada una de las tecnologías en las que está fundamentado lo que se ha hecho es confeccionar un perfil específico seleccionando determinadas características de cada uno de ellos.

Habiéndolo definido de esta forma permite que el que quiera adoptarlo encuentre que desarrollar aplicaciones es más sencillo y barato. Para la industria significa reducir al máximo el tiempo que se tarda en poner la tecnología en el mercado ya que todo lo que la compone ya está previamente desarrollado.

El estándar está compuesto por:

2.4.1 CE-HTML

CE-HTML define las funcionalidades básicas que debe soportar el navegador sobre el que se ejecutan las aplicaciones y que es el que tiene que implementar el terminal que quiera ser compatible con el estándar HbbTV.

CE-HTML es un lenguaje basado en estándares W3C. Es una definición de lenguaje HTML para dispositivos de electrónica de consumo que usa XHTML 1.0, DOM 2, CSS TV 1.0 y JavaScript. Está optimizado para la presentación de páginas HTML/JavaScript en dispositivos CE entre los que se encuentran los televisores (y teléfonos móviles, blu-rays...). De las tecnologías W3C también incorpora AJAX (XMLHttpRequest) que permite actualizar páginas web sin tener que volver a cargarlas por completo.

También incorpora un mecanismo para identificar las pulsaciones de teclas de los mandos a distancia.

2.4.2 OIPF (DAE y FORMATOS)

OIPF (Open IPTV Forum) desarrolla especificaciones para sistemas *IPTV*. Es una iniciativa con un objetivo similar al HbbTV pero en entornos de televisión *IPTV*. Han elaborado un sistema cuya definición (actualmente en su versión 2, Release 2) son 7 volúmenes de especificaciones. HbbTV incorpora partes del Volumen 2 (Media Formats) y partes del Volumen 5 (Declarative Application Environment, DAE) de la primera versión (Release 1).

Del Volumen 5 (DAE) las APIs JavaScript que son aplicables a sistemas híbridos, proporcionan funciones que permiten:

- Combinar contenido A/V lineal con páginas HTML.
- Cambiar de canal de TV o de radio.
- Añadir eventos temporizados.
- Leer metadatos y otro tipo de informaciones del TS

Del Volumen 2 se incorporan los formatos de audio, video, contenedores así como sus resoluciones, tasa de fotogramas, etc... que tiene que soportar el sistema, tanto en el canal broadband como en el canal broadcast.

2.4.3 SEÑALIZACIÓN y TRANSPORTE

Para la señalización de aplicaciones y el transporte de datos en el TS del canal broadcast se ha adherido al estándar DVB ETSI TS 102 809 v1.1.1.

El contexto de ejecución de las aplicaciones es un servicio de radio o televisión concreto dentro de un *múltiplex DVB*. La manera de señalar es la que nos encontrábamos en MHP (ambos sistemas siguen el mismo estándar para la señalización) pero se han tenido que adaptar algunas cosas. Se señala en la *AIT* (Application Information Table) del servicio DVB involucrado y se indica en su *PMT* (Program Map Table).

El estándar utiliza carruseles de objetos DSM-CC para distribuir las aplicaciones por vía broadcast. Los carruseles consisten en un árbol de directorio que se divide en una serie de módulos que pueden contener archivos y directorios/carpetas.

Los carruseles también proporcionan un mecanismo por el cual el terminal puede sincronizarse con la señal recibida en momentos determinados (intencionadamente por el radiodifusor) utilizando los llamados "Stream events".

El siguiente gráfico ayuda a visualizar la procedencia de las distintas tecnologías:

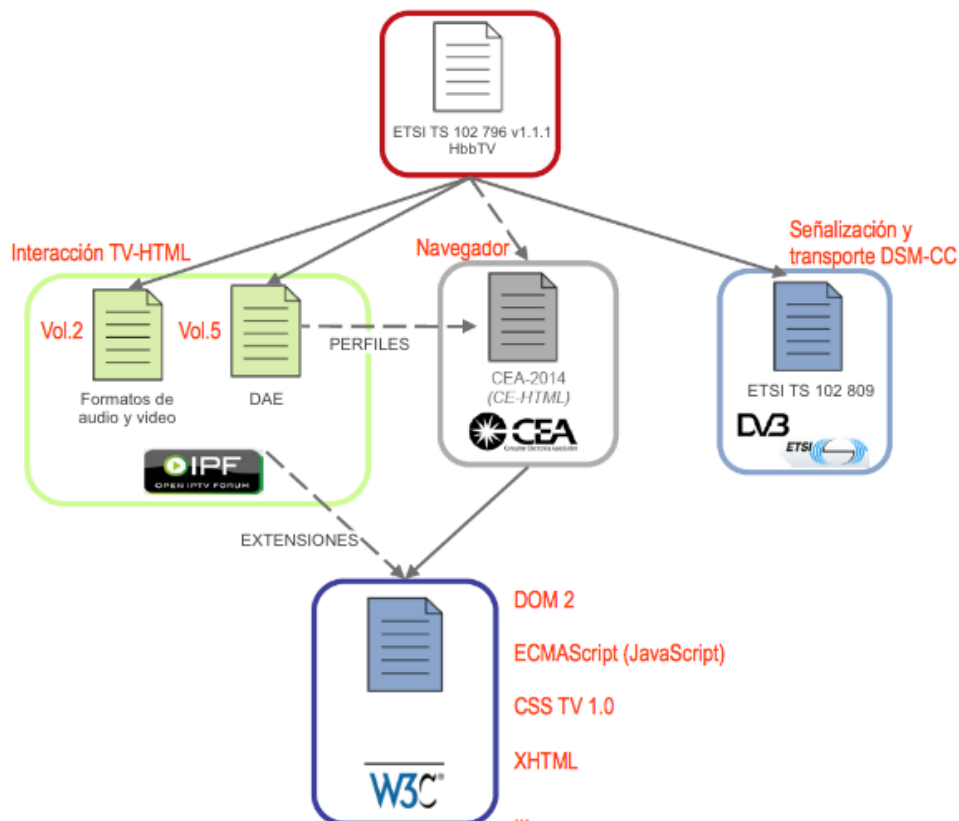


Figura 2.12 – Relaciones entre las tecnologías que componen HbbTV.

2.5 Estudios

2.5.1 Estudios de mercado

Tras dos años de existencia del estándar HbbTV en el mercado (contando desde la publicación del estándar en junio de 2010) podemos valorar el impacto que ha tenido en el mercado.

La mayor parte de los datos provienen del mercado alemán, es algo evidente ya que es donde el estándar lleva más tiempo funcionando.

Para ilustrar las previsiones de ventas en Europa primero tomamos los datos de un informe que la institución gfu (organizadores de la feria IFA en Berlín) elabora junto a ZVEI (asociación empresas de la industria tecnológica) y GfK (Investigación de mercados) hace todos los años con las previsiones en lo relativo a tendencias del mercado.

En cuanto a televisión conectada en 2011 se vendieron 6 millones de dispositivos con capacidad para internet. En las encuestas una de cada dos personas consultadas afirman que tienen previsión de comprar una TV conectada en los próximos 5 años.

La convergencia gradual que han tenido la televisión e internet en los últimos 3 años se convertirá en una tendencia de mercado dominante en 2012 y en los años sucesivos. No se trata solo de una tendencia tecnológica, se está definiendo un nuevo modelo de negocio dentro de la industria de la electrónica de consumo.

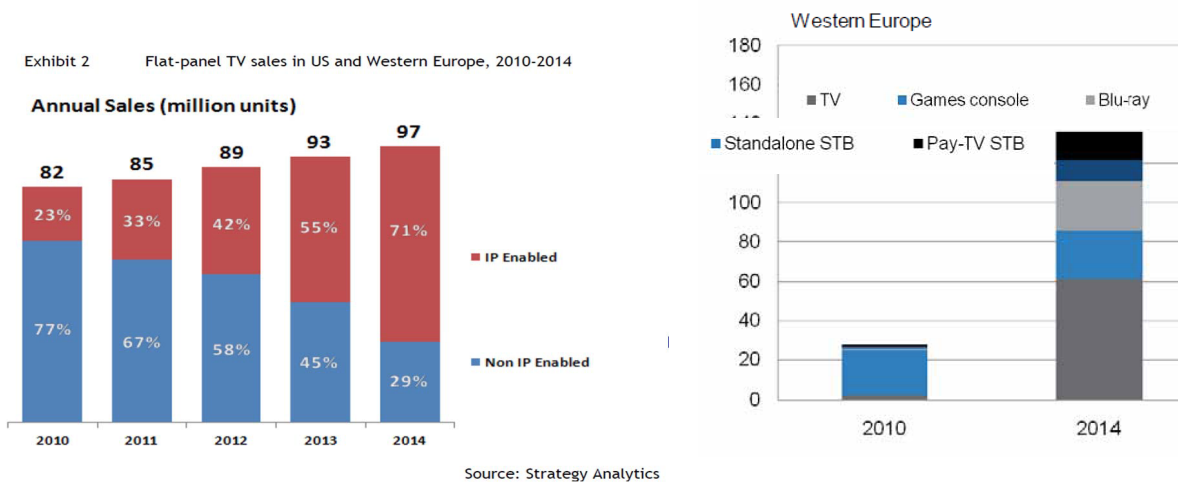


Figura 2.13 – Gráficos con perspectivas de ventas y distribución de terminales híbridos.

En el gráfico de la izquierda vemos como el 71% de la ventas de televisores en EEUU y Europa para 2014 serán TV conectadas. Aunque la realidad es que ya hoy hay algunos fabricantes que están sobrepasando esas cifras.

En el gráfico de la derecha en un análisis de Screen Digest se prevé que para 2014 el total de dispositivos conectados en Europa crezca en 80 millones de unidades. Entre estos dispositivos conectados se cuentan consolas, blu-rays, STB independientes, STB de pago y las TV conectadas que tienen un incremento de más de 50 millones de unidades.

2.5.2 Estudios de usabilidad

En Enero de 2011 la ARD hizo un estudio de usabilidad de sus aplicaciones. En el estudio participaron 45 individuos con una distribución de edades de entre 20 y 69 años.

La prueba consistió en utilizar durante una hora todas las aplicaciones HbbTV que ARD tiene disponibles (Teletexto, Mediateca (VoD), EPG, ...)

Los conceptos estudiados son:

- Tiempo de reacción de la aplicación
- Claridad
- Usabilidad
- Impresión general

Se ponderan desde “Muy malo” hasta “Muy bueno”. El gráfico de los resultados es el siguiente:

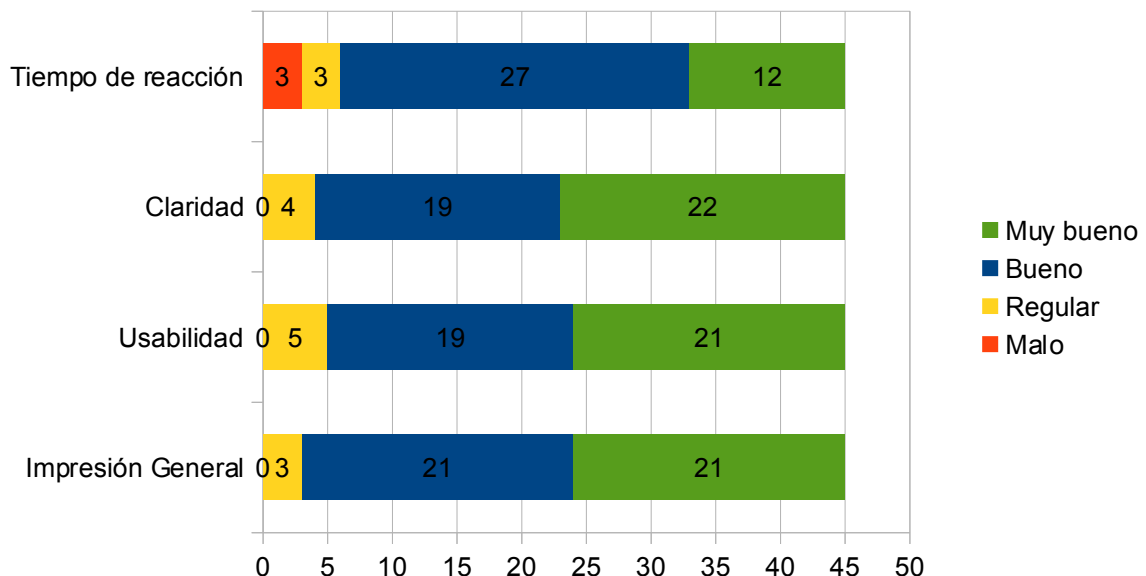


Figura 2.14 – Gráfico con representación en barras de los distintos conceptos medidos.

Se puede observar claramente que la impresión general, la usabilidad y la claridad de las aplicaciones es mayoritariamente buena o muy buena. Las aplicaciones de ARD son aplicaciones muy cuidadas y que tienen muy en cuenta que el medio sobre el que se visualizan es una pantalla de televisión. La navegación dentro de las aplicaciones es muy fluida y se pueden hacer búsquedas de video con los asistentes de teclado en pantalla de una forma muy eficaz usando solo el mando a distancia.

La nota levemente negativa viene por parte de tiempo de reacción de las aplicaciones. Puede haber varias explicaciones para esto. Mucha gente espera de una televisión conectada las prestaciones de un PC con un sintonizador de televisión y este no es el caso. La potencia de un dispositivo de CE es mucho menor que la de un ordenador personal. Por esto algunas veces las cargas de aplicaciones (sobre todo si están cargadas de funcionalidades) puede parecer algo lenta en comparación con un ordenador.



2.6 Sistemas o plataformas alternativas a HbbTV

2.6.1 Plataformas de fabricantes de televisores, STBs, etc...

En estos últimos años el gran auge de las redes de banda ancha, ha creado hábitos de consumo de vídeo y multimedia en los usuarios de ordenadores personales. Fenómenos como Youtube han sido el gran desencadenante del consumo de vídeo por internet (OTT).

Los fabricantes quieren vender más terminales y para ello están obligados a innovar. Ven una oportunidad de negocio en la conexión de la TV a internet, y por ello comienzan a fabricar terminales híbridos y para esos terminales híbridos desarrollan portales con las aplicaciones más atractivas para el usuario (sean de pago o gratuitas). Recordemos que la iniciativa HbbTV nace al amparo de los nuevos terminales híbridos desarrollados por los fabricante como parte del proceso de renovación de sus terminales. Esto ha sido un acierto porque está teniendo cierto éxito.

Pero este modelo de negocio dificulta a los radiodifusores poder estar presentes en todos los dispositivos debido a la fragmentación técnica que hay entre los portales de los distintos fabricantes. Además, es más difícil diferenciarse y se depende de un tercero para tener más o menos visibilidad.

El objetivo principal de los fabricantes a día de hoy es promocionar al máximo sus portales propietarios: Philips NetTV, LG Smart TV, Samsung Smart TV, SHARP AQUOS, Panasonic VieraCast... que cada vez incluyen más aplicaciones y servicios incorporando ya plenamente incluso las redes sociales a la par que intentar soportar cualquier tecnología que crean que les puede reportar beneficios en ventas.

Muchos de los televisores conectados que se venden ya en España son compatibles con HbbTV pero el fabricante no lo promociona ya que la adopción nacional del estándar HbbTV aun no se ha llevado a cabo. Por parte de la Administración tampoco se ha llevado a cabo ningún tipo de acción destinada a dar a conocer el sistema al usuario y por tanto es lógico que el fabricante no invierta en marketing para promocionarlo. Curiosamente en algún caso incluso viene desactivado el soporte para HbbTV y para activarlo hay que indicar a la televisión que estamos en Francia o en Alemania.

Un punto a favor de HbbTV es que todos los radiodifusores nacionales se han unido y han decidido apostar fuertemente por HbbTV. Si bien muchos de ellos ya disponen de aplicaciones para portales propietarios de fabricante, los esfuerzos en los últimos meses y en lo que tiene que venir van a ir orientados a desarrollar la tecnología que más les beneficia. Tampoco hay que olvidar que el activo más importante que tienen los radiodifusores es su propio contenido, que es el que la gente quiere ver. El radiodifusor va a centrar todos sus esfuerzos en que ese contenido sea disfrutado al máximo utilizando el estándar HbbTV.

Si los esfuerzos de los radiodifusores obtienen el resultado esperado, se promocionará el estándar ante los consumidores y entonces el fabricante podrá acometer campañas de marketing que en el mejor de los casos le reportará un aumento de ventas pudiendo así ampliar su modelo de negocio con los terminales híbridos ya que a su portal "Smart TV" se añadirían las aplicaciones HbbTV de los radiodifusores. Por poner un ejemplo, en una reunión con Philips el día 12 de julio de 2012, nos confirmaron que TODOS sus próximos modelos de televisores serán compatibles con HbbTV. Cuando hablamos de todos, nos referimos desde los de la gama más alta a los más básicos. Eso es una apuesta firme por HbbTV por parte de un fabricante.

2.6.2 Apple TV

La competencia de estas dos propuestas de TV por parte de dos gigantes del software y del hardware es más en el plano conceptual que en el plano de negocio.

Apple TV es un pequeño dispositivo sin sintonizador que permite la visualización de contenidos de audio y video exclusivamente por el canal de banda ancha e íntimamente ligado a su software iTunes.

Dicho de otra forma, es una “caja tonta” que te permite ver contenidos de iTunes en el televisor a la par que juegos y aplicaciones de las más conocidas siguiendo el mismo patrón que los portales propietarios de los fabricantes.



Figura 2.15 – Interfaz de usuario de AppleTV.

Por supuesto, la integración con cualquier otro dispositivo de la marca (iPads, iPhones, OS X, etc...) es muy buena y al tener conexión con iCloud te permite integrar el Apple TV en concepto de vida digital de la marca.



Figura 2.16 – Imagen promocional del producto y del control de un juego con un iPad.



2.6.3 GoogleTV

GoogleTV va más allá. Google ha adaptado para la pantalla de TV su sistema operativo Android y su navegador Chrome. Con ello consigue recrear el entorno de ejecución que tan bien está funcionando en los teléfonos móviles pero esta vez para ser disfrutado en una televisión.

Se pueden instalar aplicaciones (de las que el Market, ahora Google Play considere compatibles con el dispositivo) y se puede disfrutar prácticamente de cualquier contenido.

La base del sistema son otros servicios y sistemas de la marca:

- Youtube
- Chrome
- Android
- Picassa
- Google Play
- Google Music

Pero dan cabida a todo el que quiera estar incluido en sus sistema. GoogleTV se puede encontrar en STB con una única conexión a internet pero también es posible empezar a encontrarlo en televisores que tienen conexión a redes DVB, como por ejemplo el Sony internet TV:



Figura 2.17 – Sony Internet TV con Android (Tienda online Sony).

Google TV en si misma compite también con los portales de aplicaciones de los fabricantes ya que ofrecen prácticamente lo mismo en cuanto a contenidos audiovisuales, no en aplicaciones. Por eso Google está buscando la diferenciación en los contenidos ofrecidos por Youtube.

No está teniendo mucha repercusión pero Youtube está haciendo muchas inversiones para dotar de contenidos on demand y de contenidos en directo a la plataforma.

Recientemente Google ha invertido en locales en Los Ángeles para desarrollar la industria audiovisual y así poder aumentar la oferta de contenidos.

Los Angeles Becomes YouTube's Content Hub With New Playa Vista Studio



The Los Angeles area is fast becoming the content hub for online video site YouTube, a position that takes advantage of both the LA region's long history in the entertainment business, and

the rise of a slew of startups producing content specifically aimed at the YouTube audience. According to a property developer in Playa Vista, YouTube has signed an 11-year lease for a 41,000 square foot office building in the location. The new studio—which was first uncovered by the Hollywood Reporter—is in the former headquarters of Hughes Aircraft. A number of local companies are already powering YouTube's push into original content, including Los Angeles firms like Maker Studios, Machinima, Demand Media, DECA, Disney, Katalyst, My Damn Channel, and many others.

Figura 2.18 – Recorte de la noticia (Youtube).

A esto se le suma que ya es habitual que grandes cadenas de broadcast o incluso grandes productores audiovisuales, hagan la emisión en directo de sus eventos en un canal específico de Youtube:

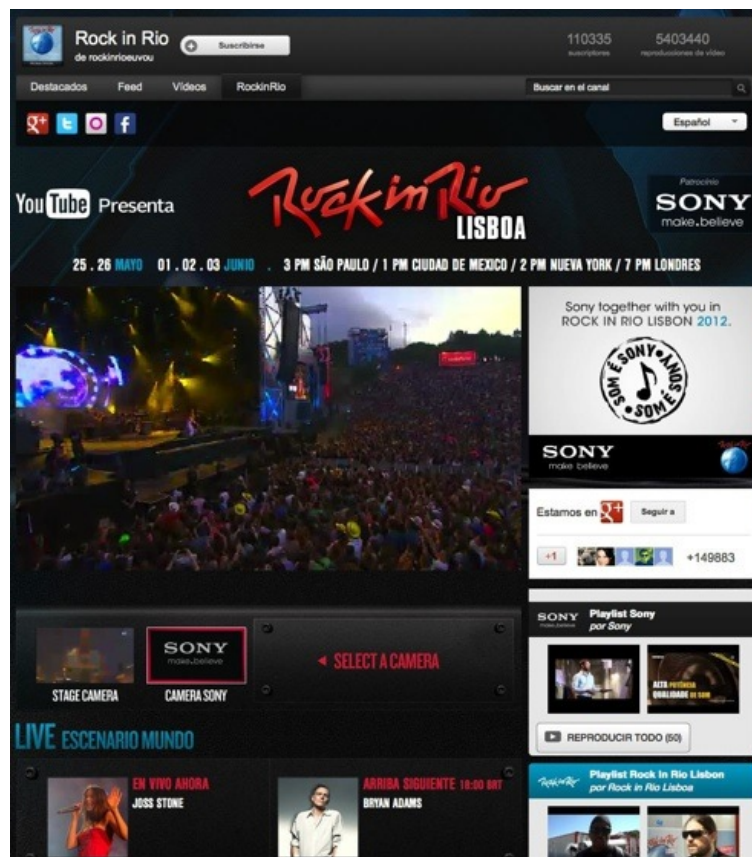


Figura 2.19 – Captura de pantalla durante un concierto en directo del festival Rock in Rio Lisboa (junio 2012).

Es decir, la aproximación de Google TV al mundo de la televisión se está fundamentando en la creación de un entorno que de libertad al espectador (Android+Chrome) en cualquier ámbito (aplicaciones, canales de VoD, video en directo) con unos contenidos cada vez más potenciados. En España, Google aún no ha formado un equipo para el desarrollo nacional del servicio.

2.6.4 Youview

Si bien todo lo que hemos visto hasta ahora son sistemas que en mayor o menor medida representan una competencia al estándar HbbTV, YouView es competencia directa aunque limitada geográficamente al territorio del Reino Unido.

Youview es el avance que se está preparando para sustituir a los actuales sistemas *Freeview* y *Freeview HD*. Youview publicó sus especificaciones técnicas en Abril de 2011 y en ella menciona a la Parte A del D-Book 7 como base de la plataforma. También menciona que usará apartados relevantes de la Parte B cuando sea publicada por el DTG. La especificación Youview es ambigua acerca de las tecnologías de presentación que debe usar o soportar. Respecto al entorno de ejecución de las aplicaciones dice “podría ser Flash Player, el motor MHEG o un navegador W3C”. Hay poca información en la especificación que permita a un desarrollador crear una aplicación compatible.

Lo llamativo del caso es que en el D-Book 7 si que se adopta HbbTV para el apartado de televisión híbrida. Le añade algunas funcionalidades (DASH, *CENC*) y requiere que sea compatible con MHEG que, recordemos, es el actual estándar para las aplicaciones interactivas de botón rojo para la televisión digital terrestre en el Reino Unido).

Esto ha creado algo de confusión en el Reino Unido, ya que el país parecía estar adoptando Youview como nuevo sistema de televisión interactiva híbrida. Youview está respaldada por la BBC, ITV, Channel 4, Channel 5, BT, TalkTalk y Arquiva y pretendían lanzar la plataforma para Febrero de 2012 pero ya ha sufrido un retraso hasta final de año (en principio)¹. Incluso Arquiva (operador de red) parece estar desembarcando también en el desarrollo de una infraestructura HbbTV con lo cual el curso de los acontecimientos en el Reino Unido aún no está claro. Tampoco se sabe si el futuro quizá sean terminales compatibles con ambos estándares. La diferencia más clara y significativa entre ambos estándares radica en los requisitos que se piden como entorno mínimo de ejecución. Mientras que en HbbTV se pide un conjunto mínimo de características que debe soportar el terminal, y deja al fabricante libertad para implementaciones, YouView requiere:

Sintonizador DVB-T2 doble
Linux 2.6.23 o superior embebido
Ruido máximo del ventilador: 26 dB
Loop RF alimentado
CPU 950 MHz
320 GB disco duro
AES128 o Triple DES para el disco duro
512 MB of RAM

...entre otras. Esto probablemente sea caro de implementar y quizá el precio del terminal no sea muy competitivo. Esto es algo que puede hacer que los fabricantes prefieran la implementación de bajo coste del HbbTV.

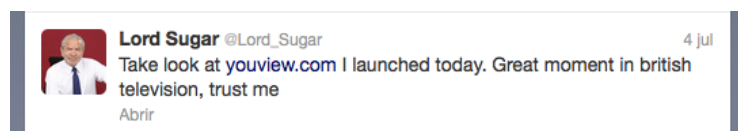


Figura 2.20 – Captura del Twitter de Alan Sugar (Amstrad) el día que presentó Youview como presidente.

1 Finalmente fue presentado el día 4 de julio de 2012. El precio del terminal está en torno a las 300 libras. No ha sido bien recibido por la crítica (<http://www.itpro.co.uk/641536/lord-sugars-youview-service-gets-thumbs-down-from-analysts>)



2.6.5 Calendario

Este es otro factor crítico para el despegue del HbbTV. En Alemania y Francia ya lo tienen fijado pero en el resto de países se debe regular y poner en marcha lo antes posible. Si se quiere que en 2012 sea el despegue definitivo, la campaña de navidad debe estar plagada de televisores con el logo del HbbTV en el marco. El tiempo juega en contra.

2.7 Despliegue (julio 2012)



Figura 2.21 – Mapa con el despliegue actual, despliegues futuros y declaración de intenciones.

Además se están haciendo pruebas en Australia, China (CCTV) y en Estados Unidos.



3. HbbTV en España

El único estándar de televisión interactiva abierto y público que había llegado a funcionar con cierto éxito en España fue MHP. Como hemos comentado en puntos anteriores de este documento, poco a poco los radiodifusores lo fueron abandonando por la falta de un parque de receptores mínimo, causado en gran medida por el pago de derechos de patentes que llevaba asociado su uso (receptor y aplicaciones) una vez lanzado al mercado.

El fracaso del lanzamiento de MHP como estándar abierto para las aplicaciones interactivas asociadas a la TDT (impulsado por la administración central con el apoyo de las CCAA que así lo exigían en sus concursos de concesiones de canales de TDT) provocó que se llegase a 2010, año del apagón analógico, y por tanto de la plena implantación de la televisión digital y que no fuese acompañado de un conjunto de servicios y aplicaciones interactivas como se había estado anunciando.

Durante 2010, pero fundamentalmente en 2011, comienzan a haber iniciativas en España con el fin de favorecer un estándar abierto y asumible por todas las partes, fijándose en la iniciativa europea e intentando adaptarlo en la medida de lo posible y conveniente a la realidad del mercado español.

Podemos destacar por un lado las iniciativas de los radiodifusores que, liderados por RTVE, se centran en “la televisión híbrida y conectada” con el objetivo de favorecer la implantación de HbbTV en España.

Paralelamente, AEDETI (Asociación Española de Empresas de Televisión interactiva), que desde 2009 había estado colaborando en el Foro Técnico de la Televisión Digital, reactiva el grupo de televisión interactiva en Abril de 2011 para promover el sector de la interactividad en España.

El “Grupo de Radiodifusores para el estudio de la televisión híbrida y conectada” arranca en junio de 2011 con la representación de RTVE, Antena 3, ETB y TV3. En Julio se une La Sexta y en Septiembre se unen al grupo Mediaset España, FORTA (Federación de Organismos de Radio y Televisión Autonómicos) y RTRM (Radiotelevisión de la Región de Murcia) a través de su dirección técnica.

Actualmente el grupo de radiodifusores está compuesto en total por 11 representantes:

RTVE, Antena 3, TVC (TV3), ETB, La Sexta, Mediaset España, RTRM (7RM), RTVV (Canal 9), Telemadrid, Net TV y FORTA.

Los objetivos principales del grupo son:

- Tener una voz única en el proceso de adopción del estándar.
- Promover el estándar
- Intentar conseguir un consenso entre todas las partes.

Se adopta como documento base el EBU-Tech 3338 “Requirements for the Standardization of Hybrid Broadcast Broadband (HBB) Television Systems and Services” (tech.ebu.ch/docs/tech/tech3338.pdf).



Ese mismo mes de junio de 2011, RTVE y Mediaset lanzan las primeras pruebas. RTVE realiza un desarrollo propio y Mediaset España lo hace con el apoyo de Telefónica. En agosto de 2011, TV3, La Sexta y algunos otros radiodifusores públicos lanzan sus servicios, todos estos con el apoyo de una plataforma desarrollada por Abertis.

Como hemos comentado, prácticamente en paralelo AEDETI reactiva el Grupo de Televisión Interactiva del Foro Técnico de la Televisión Digital por mandato de la Secretaria de Estado de Telecomunicaciones en abril de 2011.

AEDETI es una asociación de compañías de televisión interactiva. Nace en 2008 con el objetivo de promover el sector de la interactividad en España. Coordina el subgrupo de interactividad en el Foro Técnico de la TV Digital y desde su inicio se centra en la elección de un estándar para la televisión interactiva híbrida en España. El Foro Técnico de la TV Digital se crea en 2009 y en su inicio recomendó MHP al no alcanzar consenso en cuanto al estándar a elegir.

En septiembre de 2011 la SETSI publica la “Especificación de receptores de televisión digital terrestre para aplicaciones interactivas” que es elaborado por el Subgrupo de Trabajo 7 del Foro Técnico de la televisión digital, coordinado por AEDETI.

La introducción del documento indica:

En el marco del Grupo de Trabajo 7 del Foro Técnico de la Televisión Digital se ha detectado la necesidad de elaborar una serie de documentos que recojan las especificaciones mínimas que deben cumplir los receptores de televisión digital terrestre que se comercialicen en el mercado español. Para ello se ha dividido la tarea en varios subgrupos:

- *Subgrupo 1: Especificación básica de receptores de TDT.*
- *Subgrupo 2: Especificación de receptores de TDT para alta definición.*
- *Subgrupo 3: Especificación de receptores de TDT para acceso condicional.*
- *Subgrupo 4: Especificación de receptores de TDT para aplicaciones interactivas.*

Los cuatro documentos se complementan entre sí, conteniendo el primero de ellos la especificación básica que debe cumplir cualquier receptor de televisión digital terrestre que se ponga en el mercado español para garantizar plena compatibilidad con las emisiones de televisión digital terrestre y que puede complementarse con uno o varios de los documentos elaborados por los subgrupos 2, 3 y 4, dependiendo de las funcionalidades que disponga el mismo.

Este documento, elaborado por el Subgrupo 4, “Especificación de receptores de televisión digital terrestre para aplicaciones interactivas”, define los requisitos mínimos que deben cumplir los receptores para permitir la interactividad con las emisiones de televisión digital terrestre.

El documento pretende ser la adopción nacional que se hace del estándar HbbTV, y por tanto deja algunos puntos abiertos, al igual que hace el estándar a nivel paneuropeo. Estos puntos abiertos o “not agreed” (sin acuerdo) que incluye el documento son relativos a la superposición de contenido y a la elección de DRMs (mecanismos para la gestión de derechos multimedia).

De esos dos puntos, el que más conflicto crea entre radiodifusores y fabricantes tiene que ver con la superposición de contenido recibido por el canal broadband sobre el contenido recibido por el canal broadcast. El término anglosajón para esto es overlay.



El estándar europeo dice acerca del overlay que no estará permitido a no ser que:

- 1 – Haya un acuerdo entre radiodifusor y un tercero, o...
- 2 – El usuario realiza una acción voluntaria en ese sentido

Es decir, cualquiera de las dos opciones es válida.

En la “Especificación de receptores de televisión digital terrestre para aplicaciones interactivas” publicada por la *SETSI*, encontramos esta referencia en el punto REC-30.

Este punto es conflictivo porque puede alterar los modelos de negocio, sobre todo del radiodifusor. Imaginemos un partido de fútbol que se prevea que vaya a tener mucha audiencia y tras el gol de un equipo aparece un pequeña ventana en una esquina anunciando una cerveza o vendiendo la camiseta del jugador que ha marcado el gol... una ventana puesta por un tercero, no por el radiodifusor.

Se darían además ciertas paradojas legales: mientras que los radiodifusores tienen que cumplir con la Ley Audiovisual y con las restricciones de publicidad, emplazamiento de producto, horarios y/o acuerdos con las entidades que ostentan los derechos del programa, los gestores de la plataforma interactiva tendrían absoluta libertad para insertar cualquier contenido en cualquier momento.

Por tanto, la única opción que tiene el radiodifusor para proteger la integridad de sus contenidos pasa por elaborar un documento con pruebas de interoperabilidad forzando a que el fabricante cumpla para recibir un certificado que podrá exhibir para comercializar su terminal como compatible con la televisión interactiva híbrida en el mercado español.

En Mayo de 2012 la SETSI convocó a todas las partes interesadas en este proceso bajo el paraguas del subgrupo de trabajo de la televisión interactiva para la elaboración de un documento cuyo:

*“objetivo es establecer las **condiciones de implementación básicas** de la Especificación de Receptores de Televisión Digital para Aplicaciones Interactivas publicada en Septiembre del 2011 por el Foro Técnico de la TV Digital. El presente documento se incorporará como anexo a dichas especificaciones.*

El objetivo último de la interoperabilidad, en consonancia con normativa europea (ETSI, Directivas, etc.), no es otro que facilitar el desarrollo de una industria sostenible y armónica en el entorno de la televisión interactiva sin dejar de lado ningún interés legítimo. En este sentido, la televisión interactiva en Europa se está desarrollando en varios países con un modelo similar al propuesto en este documento”

Extracto de la Introducción del documento “Condiciones de implementación de la Especificación de Receptores TDT para Aplicaciones Interactivas”.



Cuando la redacción de este documento concluya, los fabricantes de terminales híbridos podrán comenzar a pasar los test de interoperabilidad y certificar sus aparatos. Será un éxito si finalmente vemos terminales híbridos con el sello de compatibilidad con la adopción nacional del estándar.

Los trabajos en el seno de este grupo de trabajo se han acelerado y comienzan a dar sus resultados, ya que la meta es activar y poner en marcha este proceso para que las primeras televisiones compatibles para el mercado español estén disponibles en la campaña de navidad de 2012 y conseguir que 2013 sea el año del despegue del HbbTV y su adopción nacional en España.

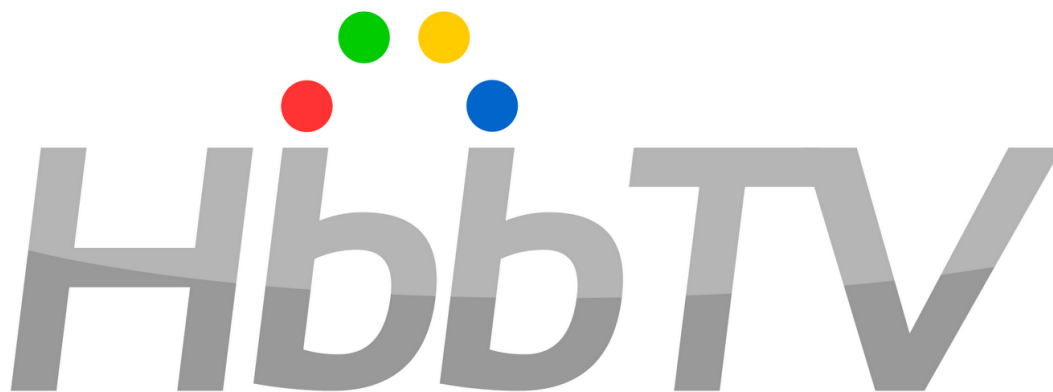


Figura 3.1 – Logo oficial de HbbTV a nivel paneuropeo.



BLOQUE 2 – Desarrollo técnico del estándar HbbTV

4. Desarrollo técnico del estándar ETSI ES 102 1796 v1.1.1

En esta parte vamos a tratar de desarrollar el estándar HbbTV 1.0 a nivel técnico: que tecnologías lo componen, que es lo que aporta cada una.

También veremos las incorporaciones más importantes que incluye la versión HbbTV 1.5 (10 de abril de 2012) que ha sido publicada por el consorcio HbbTV pero aún no ha sido publicada por la *ETSI* como estándar. También veremos que es lo que podría incorporar la futura versión 2.0, aun no publicada a fecha de este PFC.

4.1 Introducción

El estándar HbbTV define una plataforma para la señalización, transporte y presentación de aplicaciones interactivas cuyo entorno de ejecución son terminales híbridos que incluyen una conexión a un canal broadcast compatible con DVB y una conexión de banda ancha para internet.

Las funciones asociadas al canal broadcast DVB son:

- La recepción de las señales de TV, radio y servicios de datos.
- La señalización de las aplicaciones broadcast-related (bc-related, ligadas a la emisión).
- El transporte de las aplicaciones bc-related y datos asociados.
- La sincronización de las aplicaciones interactivas con los servicios de TV, radio y datos asociados.

Las funciones asociadas a la conexión a la banda ancha son:

- El transporte de contenidos bajo demanda
- El transporte de aplicaciones bc-related y bc-independent y sus datos asociados.
- El intercambio de información entre aplicaciones y servidores.
- Declaración de aplicaciones bc-independent.

Las aplicaciones se presentan al usuario en el televisor mediante un navegador HTML/Javascript cuyas funcionalidades vienen definidas en su mayor parte por los requisitos de varios de los estándares que componen la especificación HbbTV, es decir:

- CE-HTML (XHTML, DOM 2, CSS TV 1.0, EcmaScript)
- OIPF DAE
- OIPF Formatos Audio y Video



4.2 Plataforma HbbTV

En el punto 2 de este documento hemos adelantado ciertas características del HbbTV como estándar, desde el ángulo que al mercado le puede resultar interesante o ventajoso para tomar la decisión de adoptarlo o no. A nivel más funcional la plataforma HbbTV tiene las siguientes características:

- Es abierta y no depende de una sola autoridad central.
- Desde un mismo terminal se puede acceder al contenido y a los servicios de cualquier proveedor independiente.
- Las funciones básicas o “estándar” del terminal pueden ser utilizadas por cualquier aplicación pero las funcionalidades más sensibles del terminal solo pueden usarlas las aplicaciones de confianza.
- Tanto los servicios como los contenidos pueden protegerse.
- Los terminales pueden mostrar aplicaciones broadcast aunque no estén conectados al canal broadband. Independientemente de que el terminal no haya sido conectado nunca a la red broadband o de que directamente no la tenga disponible.
- Las aplicaciones pueden ejecutarse en diferentes tipos de terminal: TV conectadas, set top boxes (sintonizador híbrido externo), PVR (personal video recorder), etc...
- El estándar soporta tanto las aplicaciones bc-related como las bc-independent.

Lo que no cubre el estándar son:

- Las aplicaciones o servicios que proporciona el fabricante del dispositivo, aunque para su ejecución utilicen el navegador y las funciones que se describen en el estándar.
- Los formatos de audio, video y transporte del canal broadcast.
- Los protocolos utilizados en el canal broadcast, excepto los que estén relacionados con las aplicaciones interactivas (DSM-CC).

La plataforma combina un perfil tomado de las especificaciones del OIPF (Open IPTV Forum) con el estándar DVB para la señalización y transporte de aplicaciones interactivas en entornos *HBB*. En el estándar además, se definen los formatos multimedia soportados (para el canal broadband), el conjunto mínimo de funcionalidad del terminal y el ciclo de vida que debe seguir una aplicación.

Como ya hemos comentado anteriormente, la definición del estándar se ha hecho en base a un conjunto de características mínimo en cuanto a funcionalidades requeridas, esto hace que sea posible combinar el estándar Europeo con especificaciones nacionales que lo adapten de la mejor manera posible al territorio de adopción. Esto quizá pueda ir en detrimento de la economía de escala, pero claramente la intención es la de dar la posibilidad de reforzar los mercados nacionales.

En España se va a optar por tomar una solución intermedia. El Ministerio de Industria, Tecnología y Turismo ya ha publicado un Documento: “Especificación de receptores de televisión digital terrestre para aplicaciones interactivas” donde se detallan los requisitos mínimos que ha de tener el terminal. Estos requisitos coinciden con todos los requisitos contemplados en la versión HbbTV 1.0 más algunas características de la versión HbbTV 1.5. La parte de adopción nacional viene mayormente por lo que se va a pedir en los test de interoperabilidad.



4.3 Visión general

Tal cual está definido en el estándar, el terminal híbrido con funcionalidades web permite la descarga y ejecución de aplicaciones. Las aplicaciones están definidas como un conjunto de documentos que constituyen por si mismos un servicio interactivo o “mejorado”. Individualmente, el tipo de documentos que componen la aplicación son HTML, JavaScript, CSS, XML y archivos multimedia.

No hemos hablado hasta ahora de los servicios “mejorados” (del inglés “enhanced”). Este tipo de servicios no son servicios interactivos reales, son servicios cuya función es añadir contenido al programa o al servicio de TV/radio. Por ejemplo, un servicio mejorado sería la EPG, información de programa con imágenes. Es lo que mencionábamos como servicios interactivos locales en el punto 1 de introducción a la TVi.

Los tipos de aplicaciones que nos encontramos en HbbTV son:

1 – Broadcast-related

Son aplicaciones asociadas a uno o más programas o a uno o más eventos dentro de un programa. Decimos que pueden estar asociados a uno o más programas porque al cambiar de canal (normalmente dentro del mismo multiplex) la aplicación podría seguir activa o destruirse.

Las aplicaciones pueden arrancarse automáticamente (autostart) o pueden ser arrancadas voluntariamente por el usuario.

Las aplicaciones bc-related pueden descargarse al receptor por cualquiera de los dos canales del terminal híbrido y puede acceder a sus datos también por cualquiera de las dos vías.

2 – Broadcast-indepentent

No están asociadas a ningún servicio de broadcast. Esta aplicación descarga y accede a sus datos exclusivamente por el canal de banda ancha.

El estándar HbbTV no especifica nada en cuanto a los siguientes posibles usos del entorno del navegador compatible:

- Las aplicaciones que incorporen los proveedores de servicios (fabricantes, operadores de cable).
- El uso que se pueda hacer del navegador para ejecutar aplicaciones específicas del terminal, por ejemplo, el menú de cambio de canales, el menú de configuración del TV.
- Usar el navegador para navegar por cualquier página web.
- Que el navegador tenga implementadas otras especificaciones como *CEA-2014* o todo el paquete de especificaciones del OIPF.

4.4 Arquitectura de la plataforma HbbTV

4.4.1 Introducción.

Vamos a ver como es la arquitectura del sistema y a explicar los componentes funcionales necesarios dentro del terminal. En nivel de detalle va a ser abstracto. No se va a detallar la estructura interna de los componentes, como deben implementarse en la práctica o la forma de comunicarse entre ellos. Esto es así porque HbbTV promueve que el estándar incluya solo las características mínimas de lo que debe soportar el terminal y no promueve ni exige ninguna tecnología de implementación en concreto facilitando así la economía de escala y el time-to-market.

4.4.2 Sistema

Un terminal híbrido tiene la cualidad de poder estar conectado a dos redes en paralelo. Por un lado puede estar conectado a una red DVB (DVB-T, DVB-S o DVB-C). Por la vía del canal broadcast el terminal puede recibir emisiones A/V lineal, datos de aplicaciones y señalización de aplicaciones. Aunque no esté conectado a otra red, su conexión a la red DVB le permite recibir aplicaciones ligadas a la emisión (bc-related). También es posible enviar "Stream events" a las aplicaciones desde el canal DVB.

Si el terminal además está conectado a la red de banda ancha, entonces es posible la comunicación bidireccional con el proveedor de aplicaciones (el radiodifusor o un tercero que proporcione contenidos al radiodifusor). Por este interfaz el terminal puede recibir los datos de las aplicaciones y contenidos A/V no lineales (streaming). El terminal también podría soportar (ya que no es un requisito mínimo) la descarga de contenido A/V en tiempo no real (descarga *HTTP* no streaming).

Este es un esquema con un terminal conectado a ambas redes, siendo una de ellas la red DVB-S (Satélite):

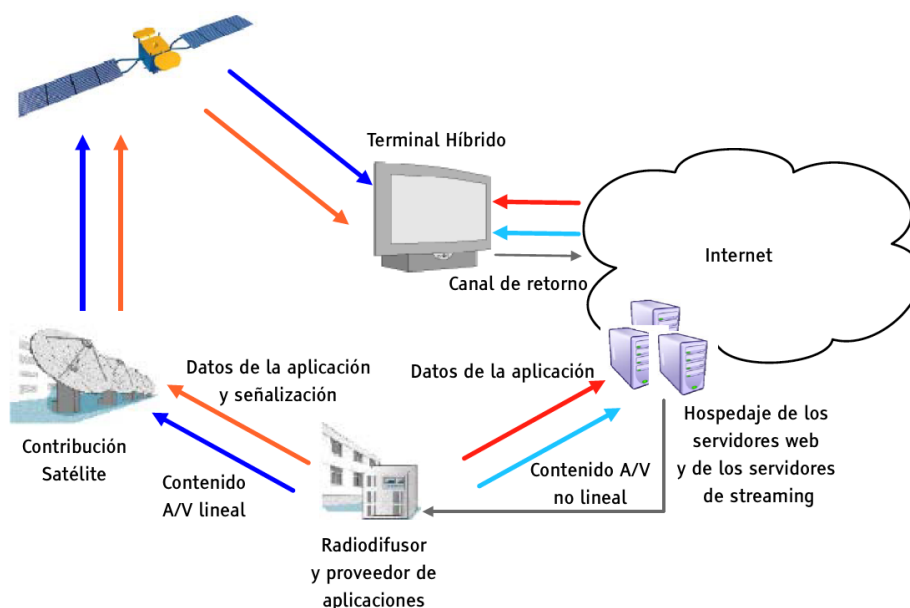


Figura 4.1 – Esquema de un sistema general HbbTV. ETSI TS 102 796 v1.1.1

4.4.3 Terminal

El siguiente esquema muestra los componentes más relevantes de un terminal híbrido:

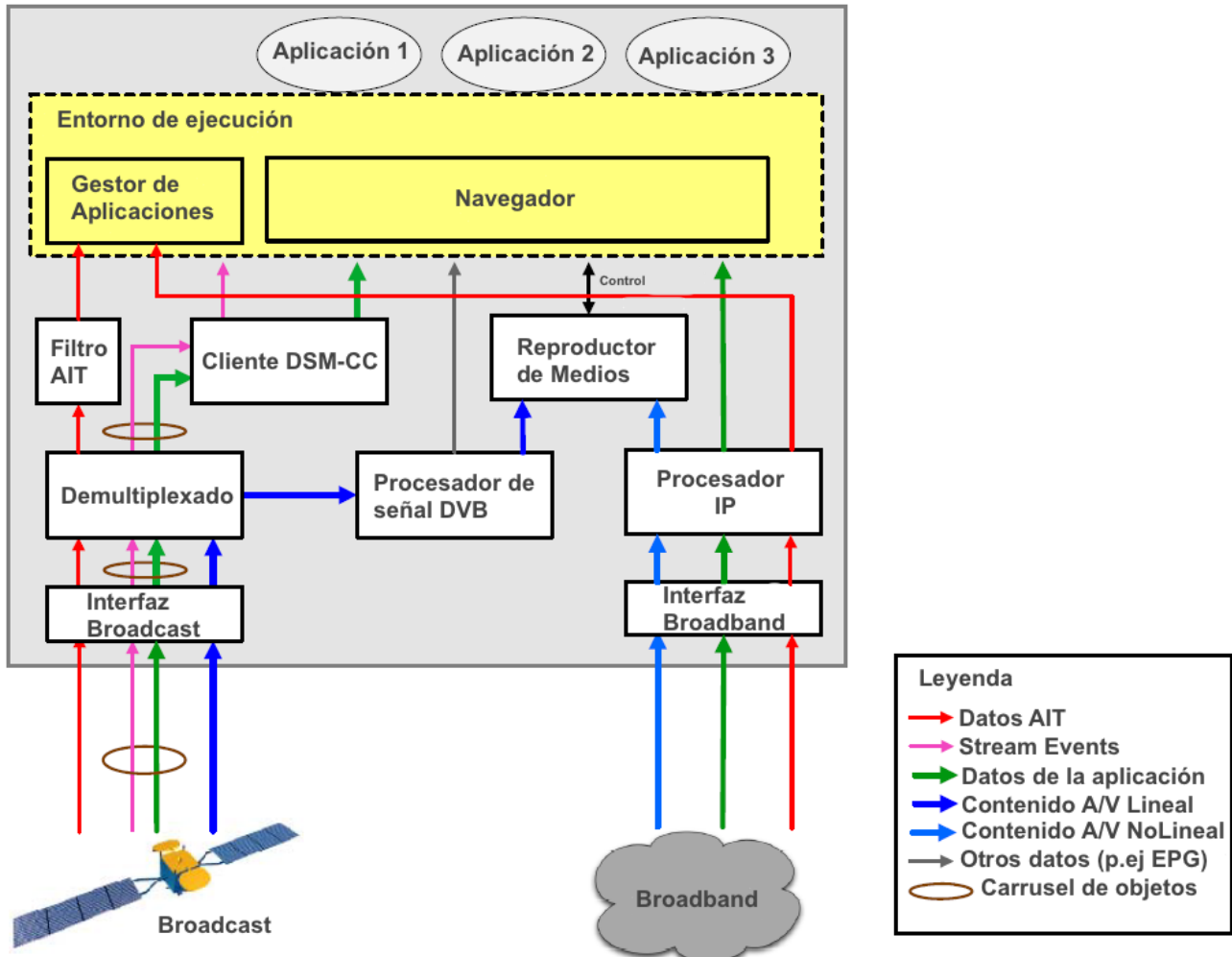


Figura 4.2 – Bloques funcionales de un terminal híbrido.

El terminal recibe datos AIT (Application Information Table), contenido TV lineal, datos de las aplicaciones y stream events por el **interfaz de broadcast**. Los datos de la aplicación y los stream events los recibe empaquetados en un carrusel de objetos DSM-CC. Por ello es necesario un **cliente DSM-CC**, para poder recuperar los datos del carrusel y entregárselos al Entorno de ejecución. Se puede entender el **entorno de ejecución** como un componente abstracto en el que se presentan y ejecutan las aplicaciones interactivas. El entorno de ejecución está compuesto por el navegador y el gestor de aplicaciones. El **gestor de aplicaciones** evalúa la AIT para poder controlar el ciclo de vida de la aplicación interactiva. **El navegador** se encarga de ejecutar y presentar las aplicaciones interactivas.

El contenido A/V lineal se procesa exactamente igual que en un terminal no híbrido. Este procesamiento se lleva a cabo en el **procesador de señal DVB** que incorpora las mismas funciones que los procesadores de un terminal no híbrido tradicional. El entorno de ejecución puede acceder a algunos datos del procesador DVB (información del canal, *EIT p/f* (EPG), y



funciones que permitan cambiar de canal). Estos datos están clasificados como “otros datos” en el esquema. Además de acceder a los datos del procesador DVB, las aplicaciones del entorno de ejecución pueden incorporar la señal A/V lineal a su interfaz y redimensionarla mediante el **reproductor de medios**. El reproductor de medios representa cualquier función relacionado con el procesamiento del contenido A/V (lineal o no).

Por el interfaz de banda ancha conectado a internet, el terminal tiene una segunda vía por la cual recibir de los servidores los datos de las aplicaciones de los proveedores de aplicaciones. Esta conexión se utiliza además para recibir contenido A/V no lineal (streaming, VoD). El **procesador IP** lleva a cabo todas las funciones necesarias para que el terminal pueda gestionar los datos que le llegan por internet. El procesador IP es quien entrega los datos de internet al entorno de ejecución. El contenido A/V no lineal recibido por internet se reenvía al Reproductor de Medios que a su vez es controlado por el entorno de ejecución Y puede hacer lo mismo que en el caso del contenido A/V lineal, es decir, incorporarlo al interfaz del programa y controlar su reproducción y redimensionarlo.

4.4.4 Capacidades mínimas y opcionales del terminal.

En cuanto a las capacidades mínimas que se exigen al terminal, el estándar incluye que debe soportar los siguientes tipos de aplicaciones interactivas:

- Aplicaciones que no usen vídeo como parte del interfaz.
- Aplicaciones que usan el vídeo DVB como parte del interfaz
- Aplicaciones que usan contenido bajo demanda en streaming unicast como parte de su interfaz. Este streaming unicast al que se refiere es al streaming HTTP, ya que el *RTSP* es opcional.

Si analizamos esto detenidamente vemos que para poder implementarlo en el terminal al menos necesitamos:

- Sintonizador DVB.
- Conexión a internet.
- Navegador capaz de mostrar aplicaciones interactivas.
- Que el navegador tenga capacidad para mostrar y redimensionar el video DVB.
- Que el navegador tenga capacidad de reproducir streaming de contenido de internet.

Esta es la base que HbbTV determina que debe incorporar un terminal híbrido.

Opcionalmente, hay otras funciones que pueden ser soportadas por el terminal:

- La descarga de contenido A/V en una memoria local en el terminal. Se podrá descargar contenido (un archivo de vídeo, podcast,...) y contenido progresivo (un streaming de video). Esta función es denominada “función de descarga”.
- La programación de grabaciones y su reproducción y el “timeshifting” del contenido lineal DVB usando el almacenamiento que tenga disponible el terminal (interno o externo). Esta función es denominada “función PVR”.
- Soporte para streaming usando los protocolos RTSP / RTP. Esta función es denominada “función RTSP”.
- El contenido protegido por el interfaz de banda ancha.

4.5 Fundamentos tecnológicos del estándar HbbTV

HbbTV se basa en los siguientes estándares/especificaciones:

- CEA-2014 A
- OIPF DAE y Media Formats
- ETSI TS 102 809

De cada una de ellos HbbTV toma ciertas funcionalidades y perfila (particulariza) ciertos elementos para que sean específicos para HbbTV.

Vamos a hacer una breve introducción de cada uno de los estándares/especificaciones individualmente para ver que es lo que HbbTV incorpora de cada uno de ellos.

4.5.1 CEA-2014 A

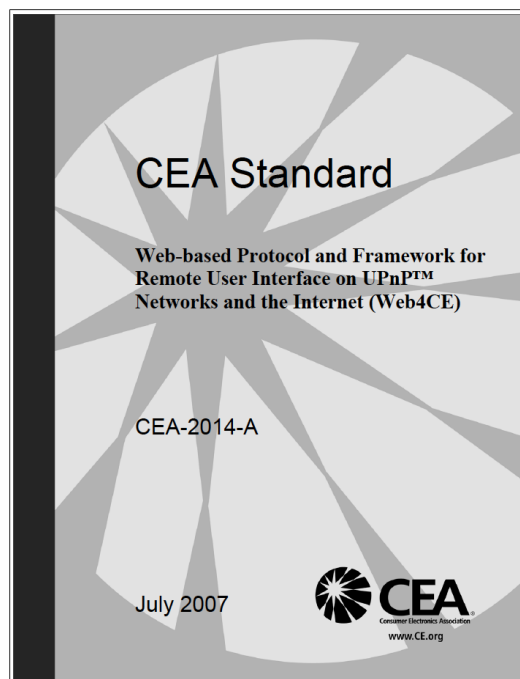


Figura 4.3 – Portada del estándar CEA-2014-A

El estándar ANSI/CEA-2014-A “Web-based Protocol and Framework for Remote User Interface on *UPnP* Networks and the Internet” (también conocido como *Web4CE*, de junio de 2006) define los mecanismos necesarios para permitir que un interfaz de usuario pueda ser presentado y controlado de forma remota y por dispositivos o puntos de control distintos al que hace la presentación del interfaz.

Las operaciones básicas que puede hacer el dispositivo toman como referencia otro estándar, el *UPnP* v1.0 para redes domésticas e internet. El estándar CEA-2014 también permite que sean terceros de internet los que provean los interfaces a mostrar en dispositivos del hogar y provee de funciones para TV, teléfonos móviles y dispositivos portátiles.

También incluye mecanismos cuya intención es la de habilitar el flujo de contenidos que pueden estar protegidos por DRMs.

Los principales objetivos del CEA-2014 son:

- Proveer un mecanismo que permita la presentación y control remoto de interfaces de usuario dirigidos a dispositivos de electrónica de consumo.
- Controlar los contenidos del interfaz de usuario que pueden estar en un servidor UPnP (en el hogar) o en internet.
- Usar estándares web existentes para la presentación del contenido del interfaz de usuario y extenderlos solo cuando sea necesario (premisa presente también en el estándar HbbTV).
- Estar disponible para la mayor variedad de dispositivos CE posible: TV, STBs, móviles, consolas,....
- Permitir la comunicación dinámica entre interfaces remotos y servidores, esto incluye permitir actualizaciones parciales del interfaz en determinados intervalos de tiempo.
- Permitir que el usuario reciba notificaciones por parte del servidor en cualquier momento (si el usuario lo permite).
- Permitir que contenido A/V forme parte del interfaz.

El énfasis en la presentación remota de interfaces se explica por la intención práctica de este estándar en que cualquier dispositivo CE pudiera representar un contenido, interrumpirlo y continuar la visualización del mismo en otro distinto (TV → móvil, por ejemplo), que las sesiones se suspendiesen en un terminal y se retomaran en otro, que un terminal pudiera ser servidor de otro que hiciese de cliente y que todos los terminales pudiesen estar conectados a internet.

Por tanto CEA-2014 básicamente define 4 cosas:

- Dispositivos: Clientes y servidores de interfaces de usuario remotos.
- La interacción entre el cliente y el servidor en la red doméstica (UPnP).
- La interacción entre el cliente y el servidor en internet.
- Un nuevo tipo de navegador que sería requisito en los dispositivos compatibles.

Todas estas pretensiones de interacción entre distintos dispositivos de tan distinta naturaleza (TV, móvil, consola,...) necesitan de tecnologías capaces de ser aplicadas en distintos dominios, por ejemplo, el dominio del PC, del móvil, de la televisión, el hogar e internet.

Un ejemplo que sirve para entender un poco mejor lo que pretende el estándar mediante la definición de interfaces remotos, tecnologías “multi-ámbito” y comunicaciones entre clientes y servidores es el siguiente:

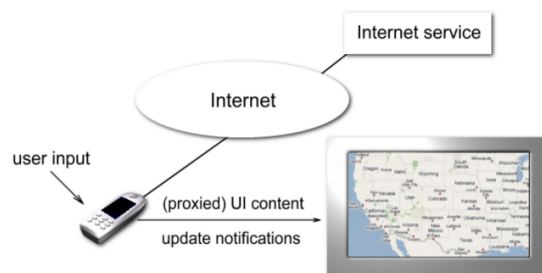


Figura 4.4 – Representación de la pantalla de un móvil remotamente en la pantalla de un televisor.

Web4CE: Accessing Web-based Applications on Consumer Devices. Dee / Shrubsole (Philips Research Laboratories)



En él se explica como un usuario que está usando el teléfono móvil para consultar una ruta de Google Maps, al tener cerca una TV con una pantalla grande, la usa como pantalla para la navegación en vez de usar la del teléfono, es decir, el teléfono se convierte en servidor de interfaz de la televisión que sería el cliente. Web4CE sería el Framework capaz de habilitar esta interacción entre dispositivos. Web4CE adopta una gran variedad de tecnologías en el ámbito doméstico y de internet que son capaces de funcionar en distintos dominios. Estas tecnologías son:

- XHTML
- DOM Level 2
- CSS TV Profile 1.0 (Es una ampliación del perfil CSS Mobile Profile)
- UPnP
- AJAX (XMLHttpRequest)
- ECMAScript 262 3ª Edición (JavaScript estandarizado por ECMA)

Con Web4CE se definió también el lenguaje CE-HTML, que no es más que el nombre que se da a la colección de tecnologías anteriores formada por:

- XHTML
- CSS TV
- Componente XMLHttpRequest
- DOM Level 2 (Core, Style, Events, HTML)
- ECMAScript

Para poder funcionar con toda la variedad de dispositivos a los que está dirigido Web4CE es necesario extenderlo para añadirle ciertas funciones. Una de ellas es la conocida como "Intercambio de capacidades". Con este mecanismo un servidor puede adaptar el interfaz de usuario a las capacidades del dispositivo (bien sea un móvil o una consola). Web4CE define un mecanismo basado en unos perfiles abstractos:

- "SD_UIPROF" (predeterminado), que describe clientes con una pantalla de 640x480 píxeles, un conjunto básico de teclas y unos tipos de letra que permiten su lectura para distintos tamaños y a varias distancias.
- "HD_UIPROF" para pantallas de 1280x720 píxeles.
- "MD_UIPORF", que es utilizado para dispositivos de gama baja, con una resolución de al menos 208x208, 4096 colores y menos teclas.

Web4CE define un lenguaje XML para poder intercambiar capacidades por encima de las anteriores y para que los servidores listen las capacidades requeridas para que pueda llevarse a cabo la conexión. Otra parte muy importante de CEA-2014 (Web4CE) es el soporte que se da para mostrar aplicaciones en la televisión y poder interactuar con ellas usando el mando a distancia. Por tanto, Web4CE define las extensiones necesarias para poder gestionar las pulsaciones de teclas remotas del mando a distancia. Para ello introduce unas etiquetas llamadas `<op>` que permiten mapear las teclas del mando con unos códigos de tecla del mando a distancia o del teléfono móvil. De esta forma el desarrollador puede meter en su aplicación una línea del tipo *"Por favor, presione el botón `<op code="VK_RED">` para arrancar la aplicación"* sin tener que preocuparse de como esté identificado el botón en el mando a distancia (aunque el del ejemplo no da lugar a error, es el botón rojo).



Para permitir la comunicación bidireccional entre cliente/servidor y las actualizaciones parciales del interfaz se usa el mecanismo XMLHttpRequest (AJAX) que es el estándar de facto para este tipo de operaciones y el detonante de la llamada web 2.0. El XMLHttpRequest es un método que hace consultas periódicas al servidor para recibir nuevo contenido y mostrarlo.

No obstante, para algunos usos podría suponer un retardo esperar a que el cliente consulte al servidor usando el método XMLHttpRequest, y por ello se crea un objeto de scripting denominado "NotifSocket". NotifSocket permite crear una conexión TCP persistente entre cliente y servidor, estableciendo así un canal permanente por el cual el servidor puede avisar o notificar en cualquier momento de cualquier cambio en el interfaz sin tener que esperar el tiempo entre consultas programadas del método XMLHttpRequest. Un ejemplo de uso de NotifSocket sería el aviso en el televisor de la identidad de la persona que en ese momento está llamando por teléfono.

Web4CE también proporciona mecanismos para poder acceder a contenido A/V e integrarlo en el interfaz de usuario. Permite los canales alfa (combinación de colores en los que se permiten las transparencias) y la superposición de partes del interfaz de usuario sobre el video, el control de la reproducción de un flujo A/V y controlar el tamaño de la ventana de video. También permite que el contenido A/V se mantenga entre distintas páginas del interfaz.

De todo lo anterior, HbbTV incorpora lo siguiente:

- Un perfil de CE-HTML (XHTML 1.0, DOM 2.0, CSS TV, XMLHttpRequest) que va a ser el lenguaje usado en las aplicaciones.
- Definición de los códigos de las teclas (los códigos no directos, los virtuales VK_*) y su acceso mediante el DOM.
- Capacidad de incrustar contenido A/V no lineal en las aplicaciones.
- Especificaciones para los formatos de imagen fija.
- Toma la idea de los perfiles de acceso y el intercambio de capacidades.

El estándar CEA-2014 no es una referencia normativa dentro del estándar HbbTV, es una referencia informativa y por tanto, según el estándar HbbTV, el documento CEA-2014-A no es necesario para la aplicación del estándar.

Todo lo que se adopta del estándar CEA-2014 (directa o indirectamente) se hace a través de las especificaciones del OIPF, que a su vez modifica (hace un perfil) el CEA-2014 para su uso específico en los entornos IPTV.

La realidad es que sin el documento CEA-2014 no es posible saber exactamente que se incluye en el lenguaje CE-HTML ya que el perfil CE-HTML del DAE hace constantes referencias al CEA-2014 y bastantes modificaciones, pero no lista explícitamente el resultado de aplicar esas modificaciones al CEA-2014

Por mi experiencia personal en el desarrollo de aplicaciones en HbbTV, considero bastante interesante disponer de una copia de este estándar. No es imprescindible, pero ayuda bastante al desarrollo el poder tener una referencia directa del lenguaje CE-HTML.

Vamos a hacer un esquema que sirva de guía para poder trazar las modificaciones que ha sufrido CE-HTML desde su origen en CEA-2014 hasta su perfil HbbTV, sin bajar al detalle.

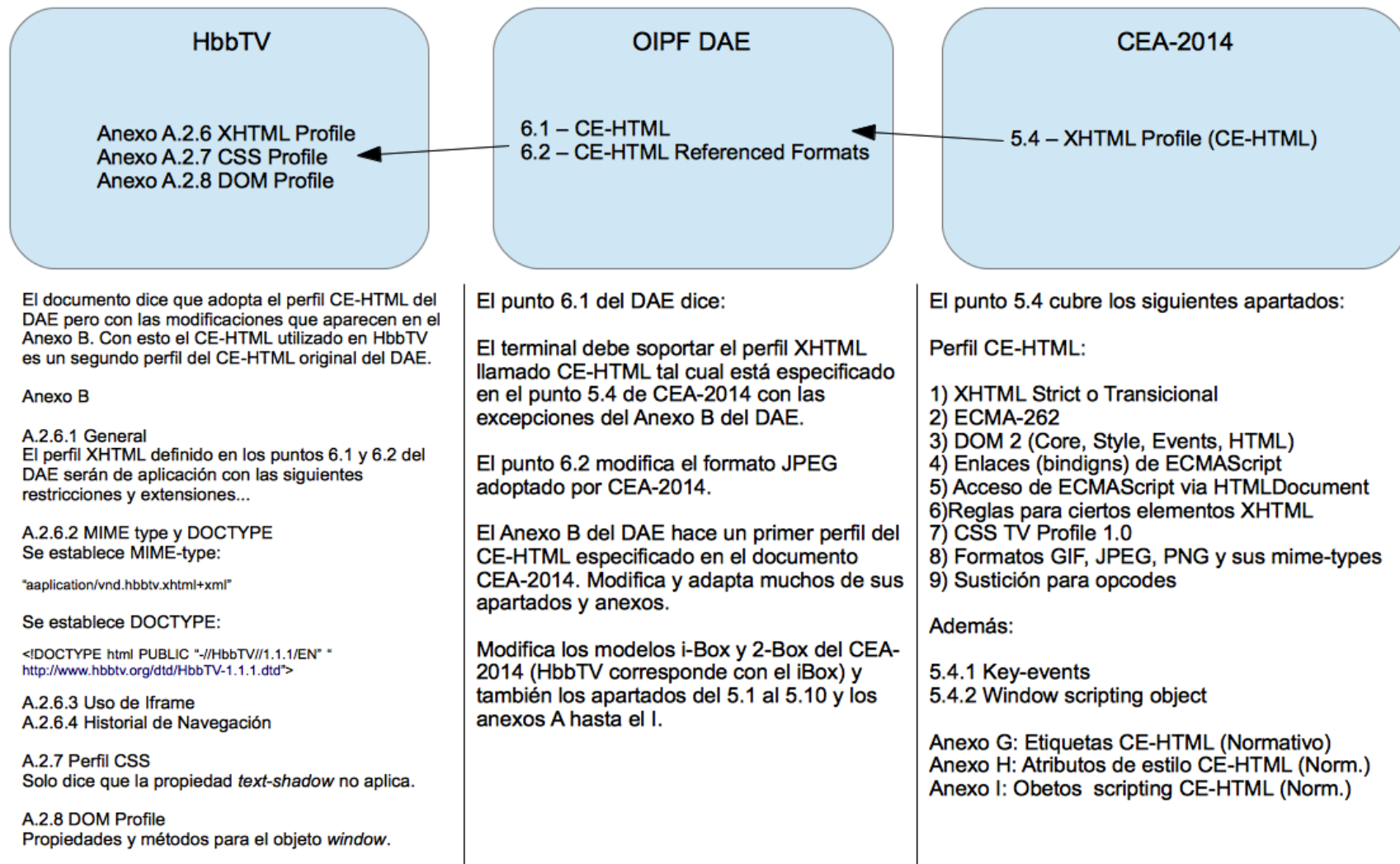


Figura 4.5 – Trazabilidad para el lenguaje CE-HTML.



Vista la evolución que sigue CE-HTML hasta llegar al estándar HbbTV, antes de avanzar en el desarrollo de las especificaciones del OIPF, vamos a hacer un breve repaso de las tecnologías W3C que componen CE-HTML, y por tanto la base del lenguaje que se va a utilizar para la creación de las aplicaciones HbbTV, estos son:

- XHTML 1.0 Strict o Transitional
- DOM 2.0 (Core, Style, Events + KeyEvents (teclas VK_*) + DOM2 HTML)
- CSS TV Profile1.0 (Basado en CSS1 + elementos CSS 2.0 y CSS2.1)
- XMLHttpRequest

4.5.1.1 XHTML 1.0 Strict o Transitional

XHTML es una familia de tipos de documento que reproducen, reducen y extienden el lenguaje *HTML4*. XHTML está basado en documentos XML (eXtensible Markup Language) y está pensado para funcionar con *user-agent* tipo XML. XML es una mejora de *SGML* (en el cual está basado HTML4). XHTML 1.0 es el primer tipo de documento de la familia XHTML y es una reformulación de HTML4 en base a documentos tipo XML. Es posible hacer que código XHTML funcione en navegadores compatibles con HTML4 con solo seguir unas pequeñas pautas.

¿Qué ventajas tiene XHTML sobre HTML4 para ser el elegido?

- Cualquier nueva necesidad es fácil de incorporarla a XHTML mediante módulos que permiten crear contenido con lo anterior y con lo nuevo.
- XHTML se ajusta a XML y por tanto es fácil de validar, leer y editar.
- Los documentos XHTML se pueden crear de forma que funcionen igual o mejor que los documentos originales HTML4.
- Los documentos XHTML pueden usar aplicaciones (p.ej scripts) que estén basados tanto en el DOM HTML como en el DOM XML (en el siguiente apartado vemos el DOM). En ese aspecto es "híbrido" como el HbbTV.
- Desarrollar en XHTML es beneficioso a largo plazo porque los documentos generados son muy portables.

Desde el W3C apuestan por XHTML como el lenguaje para generar los contenidos Internet, aunque está costando que sea ampliamente adoptado ya que XHTML requiere de más precisión que HTML4 a la hora de desarrollar. HTML permite imprecisiones en el código que el *parser* corrige o pasa por alto, en XHTML un fallo da error.

Terminología XHTML que vamos a definir muy brevemente:

- *DTD* (Document Type Definition): Es una colección de declaraciones XML que, en su conjunto, definen la estructura "legal" de elementos y atributos disponibles en un documento que se quiera ajustar a ese DTD.
- Elemento: Es una unidad estructural del documento, definido en el DTD.
- Atributo: Es un parámetro de un elemento. Su tipo y valor están declarados en el DTD.
- User-agent: Es el sistema que procesa el documento de acuerdo a la especificación.
- Renderizar: El proceso de presentar la información contenida en un documento.

4.5.1.2 DOM 2.0

El Document Object Model es un interfaz (independiente de plataforma y lenguaje) que permite a los programas y scripts acceder, actualizar y modificar el contenido y la estructura de un documento. El DOM 2.0 Core está hecho en base a un conjunto fundamental de interfaces que permite la creación y manipulación de la estructura y contenidos de un documento.

El DOM internamente organiza y hace referencia a los elementos siguiendo una estructura de nodos organizados jerárquicamente en forma de árbol. Este es un DOM XML, que representa a lo que sería una página web XHTML:

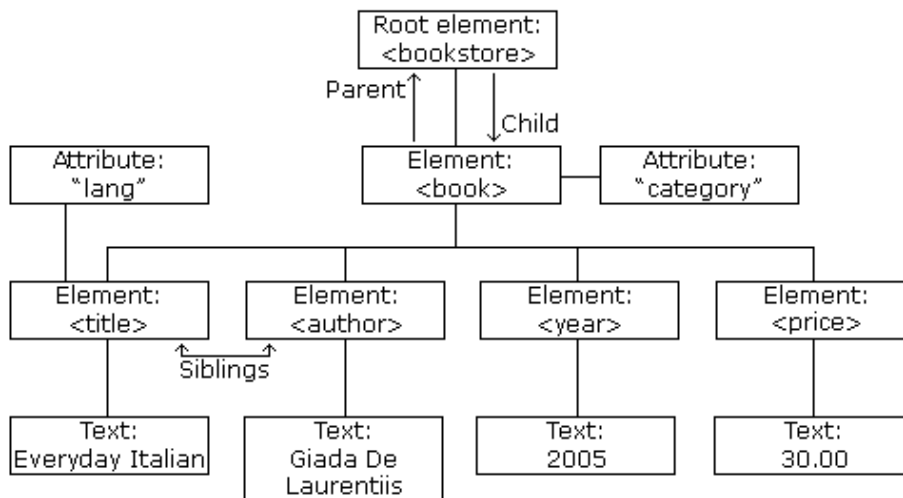


Figura 4.6 – Bloques funcionales de un terminal híbrido.

<http://www.w3schools.com/dom/default.asp>

Como decimos, el DOM es un interfaz no ligado a plataforma o lenguaje. Pero para poder usarlo es necesario que haya algún lenguaje que a modo de “vehículo” lo implemente y permita usarlo en un documento, esto se llama “language binding”. En el caso del DOM, el “language binding” que vamos a manejar está hecho con JavaScript.

Si el DOM Core permite el acceso de programas y scripts al interior de los documentos para modificarlos o actualizarlos, el DOM Style hace lo mismo pero para los documentos CSS (Cascade Style Sheets, Hojas de estilo en cascada, los vemos más adelante). Y el DOM Events, que también está construido sobre el DOM Core, tiene 2 objetivos. El primero es diseñar un sistema de eventos genérico que permita el registro de manejadores de eventos (event handler) y el segundo es crear un subconjunto común para usar con DOM 0, esto es para conseguir retrocompatibilidad.

4.5.1.3 XMLHttpRequest

La especificación XMLHttpRequest define un API que permite que un cliente pueda transferir datos entre un servidor y un cliente mediante scripting, todo esto hecho en segundo plano. Permite al desarrollador actualizar una página web sin tener que recargarla por completo, seguir pidiendo datos al servidor tras haber cargado la página, recibir datos del servidor (sin tener que pedirlos) tras la carga de la página y seguir mandando datos al servidor en segundo plano. XMLHttpRequest forma parte del API HTTP ECMAScript.



4.5.1.4 CSS TV Profile 1.0

Esta especificación es un subconjunto de CSS2 y del módulo Color de CSS3, especialmente indicada para las necesidades de televisiones, STBs y otros dispositivos cuyo fin es ser visualizados en una pantalla de TV.

CSS2 es un lenguaje de hojas de estilo en cascada que permite a los desarrolladores dotar de “estilo” (fuentes, espaciado, ...) a documentos estructurados (HTML y aplicaciones XML). Su fundamento consiste en separar el estilo de presentación del contenido del documento, así se consigue simplificar el desarrollo y mantenimiento de las aplicaciones web. CSS2 permite hojas de estilo específicas para distintos medios (impresoras, móviles, dispositivos Braille,...). El perfil CSS TV conforma un perfil específico para televisiones, identificando un conjunto mínimo de propiedades, valores, selectores y reglas de anidamiento.

Para formar CSS TV se ha utilizado una gran parte de CSS1, partes de CSS2 y el módulo Color de CSS3. En sí, CSS TV Profile es una ampliación del CSS Perfil Móviles, en el cual dentro del código se usa el tipo `'tv'` en vez del tipo `'handheld'`.

Separar el documento de su hoja de estilos quiere decir que la hoja CSS está en un documento aparte con extensión `.css` y que tiene que ser accesible por el documento principal.

En el caso de HbbTV el CSS además se puede utilizar para gestionar la navegación por la pantalla.

Queriendo hacer un poco más de hincapié en las funciones que pueden ser definidas con una hoja de estilos, cabría indicar que para desarrollar aplicaciones que van a ser presentadas en una pantalla de televisión y que van a ser comandadas generalmente por un mando a distancia, es muy importante tener en cuenta la usabilidad de la aplicación.

Esto significa que hay que conseguir que la navegación dentro de la aplicación sea cómoda. Teniendo en cuenta que hay una cierta distancia entre la pantalla y el usuario y que el usuario dispondrá de 4 flechas y un OK para poder hacerlo todo es muy importante diseñar los portales de tal forma que con pocos movimientos se llegue al lugar deseado y que en el trayecto hasta el lugar deseado quede claro donde nos ubicamos en cada momento. Al no tener un ratón no nos ponemos directamente donde queremos, tener que llegar hasta ahí y si en algún momento nos perdemos no sabremos llegar.

Es por esto que muchas veces se dice que una página HbbTV no es lo mismo que una página web para la televisión.

Todo esto lo veremos en detalle un poco más adelante, pero era apropiado indicarlo aquí, ya que la usabilidad de la aplicación se consigue trabajando con las hojas de estilo CSS.



4.5.2 OIPF

El OIPF (Open IPTV Forum) es una organización cuya misión es la de acelerar la creación de un mercado para el IPTV definiendo y publicando especificaciones gratuitas basadas en estándares para los servicios IPTV extremo a extremo del futuro. El que las soluciones estén orientadas extremo a extremo (especificando en toda la cadena) es positivo para la interoperabilidad, facilidad de conexión y experiencia final del usuario.

Hasta hoy han publicado 2 versiones de sus especificaciones que son las siguientes:

Specifications Release 2

- [Service and Platform Requirements V2.0 \(PDF\)](#) - 2008/12/12
- [Services and Functions for Release 2 V1.0 \(PDF\)](#) - 2008/10/20
- [Functional Architecture V2.1 \(PDF 1.65 MB\)](#) - 2011/03/15
- [NEW Solution Specification Volume 1 - Overview V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 2 - Media Formats V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 2a - HTTP Adaptive Streaming V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 3 - Content Metadata V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 4 - Protocols V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 4a - Examples of IPTV Protocol Sequences V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 5 - Declarative Application Environment V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 6 - Procedural Application Environment V2.1 \(PDF\)](#) - 2011-06-21
- [NEW Solution Specification Volume 7 - Authentication, Content Protection and Service Protection V2.1 \(PDF\)](#) - 2011-06-21
- [NEW V2.1 XML Schemas \(ZIP\)](#) - 2011-05-20

Figura 4.7 – Captura de la web del OIPF.

<http://www.oipf.tv/specifications-release-2>

HbbTV se basa en la versión 1 de las especificaciones. Concretamente toma como referencia bastantes partes del Volumen 5 – Declarative Application Environment y el Volumen 2 – Media Formats.

4.5.2.1 OIPF DAE

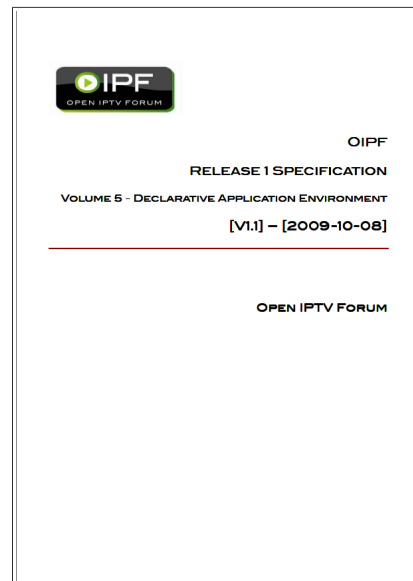


Figura 4.8 – Portada de la especificación OIPF Vol. 5 - DAE.

El motivo de incorporar parte de esta especificación al estándar HbbTV es que el CE-HTML original no ofrece ningún interfaz que permita acceder a contenidos DVB. El perfil del navegador del OIPF DAE sí que contempla esta posibilidad. Al ser una especificación desarrollada para sistemas IPTV es factible aplicarlos en entornos híbridos.

Los APIs JavaScript que HbbTV selecciona del DAE permiten:

- Integrar señales DVB en la aplicación
- Sintonizar otra emisora de TV o de radio desde la aplicación
- Leer metadatos de la señal DVB
- Control PVR (si se desea implementar, ya que es una función opcional del estándar HbbTV)

Los componentes seleccionados de CE-HTML y del OIPF definen las funciones principales del navegador HbbTV. Incorpora partes completas del OIPF DAE, modifica otras y especifica algunas nuevas. El resultado, HbbTV lo llama “Browser application environment” (BAE). Este es un resumen de sus funcionalidades más representativas:

Funciones y APIs derivadas del OIPF DAE

Aplicación

- Solo se permite una sola aplicación en ejecución y no permite aplicación en segundo plano.
- Concepto de aplicación de confianza y aplicación de no confianza.



Gestión de recursos

- Ciclo de vida de la aplicación.
- Al crear una instancia de un objeto (excluidos los objetos video/broadcast) se puede indicar que existe una falta de recursos.
- Control parental.

Descarga de contenidos

- Gestor de descargas (si está implementado).

Streaming

- Streaming unicast por HTTP.

Contenidos programados (programas de televisión lineales)

- Lectura de la lista de canales y además de la lista de grabaciones programadas (si está implementada la opción de PVR).

Ciclo de vida de la aplicación

- Soporta aplicaciones web normales pero en el marco de uso de las aplicaciones bc-independent.
- Se puede crear aplicaciones desde la aplicación en ejecución.
- No admite el método NotifSocket.
- El agente DRM (si está implementado) puede arrancar una aplicación.

Notificación de eventos

- Está implementado el XMLHttpRequest (no el NotifSocket ni el servicio de llamadas)

APIs para la gestión de aplicaciones

- Métodos del gestor de aplicaciones (**Application Manager**)
- Hay una clase "application" que controla los estados de la aplicación (*show()*, *hide()*, *destroyApplication()*,...)
- Hay una clase "ApplicationPrivateData" con propiedades del tipo *keyset*, *currentChannel*.

APIs para la configuración y los ajustes

- Objeto *application/oipfConfiguration* para saber las características que ofrece el dispositivo.
- Clase "Configuration" con propiedades del tipo: *preferredAudioLanguage*, *preferredSubtitleLanguage*, *countryID*...



APIs para la descarga de contenidos

- Objeto `application/oipfDownloadTrigger` solo para aplicaciones de confianza y si la funcionalidad de descarga está implementada en el aparato.
- Objeto `application/oipfDownloadManager` (igual que el anterior).

APIs de control parental

- Objeto `application/oipfParentalControl`, y clases `ParentalRatingScheme`, `ParentalRating`.

APIs para la grabación de contenidos A/V lineales

- Objeto `application/oipfRecordingScheduler`, solo para aplicaciones de confianza y si la función PVR está implementada. La clase `Recording` tiene las siguientes propiedades: `state`, `id`, `recordingStartTime`, `recordingDuration`, `parentalRatings`. La funcionalidad PVR en primeras fases de despliegue del HbbTV no se va a utilizar, al menos por parte del radiodifusor. Algunos radiodifusores se lo han planteado pero al final han decidido no implementarlo.

APIs para metadatos

- No hay ninguna implementada del OIPF DAE. No significa que no se puedan hacer búsquedas de datos, lo que quiere decir es que no se va a poder hacer con un objeto integrado del Browser de HbbTV.

Video A/V lineal (señal TV lineal)

- Objeto `video/broadcast` con métodos como: `setChannel()`, `nextChannel()`, `prevChannel()`, `bindToCurrentChannel()`, `release()` y `stop()`. Las aplicaciones bc-related tienen siempre acceso y control sobre el objeto `video/broadcast`. Las aplicaciones bc-independent tienen acceso con condiciones y restricciones.
- Hay extensiones para la grabación y el timeshift (si está implementado el PVR).
- Existe acceso al DVB-SI p/f (Service Information previous/following para el EPG).
- Tiene una clase `ChannelList` que permite leer la lista de canales e ir a uno de ellos.

Video A/V no lineal (Streaming)

- Hay objetos que permiten la reproducción de contenido streaming, su descarga (si está implementada la función, para reproducir contenido grabado, *trickmodes* (avance, retroceso pausa, cambio de velocidad de reproducción del video streaming).
- Eventos DOM2 para los objetos A/V.
- Reproducción de audio desde memoria.



APIs varias

- Objeto *application/oipfCapabilities*. Sirve para que un servidor pueda preguntar a un terminal (o el terminal mandar información) de las características que tiene implementado. Así se puede saber si el terminal soporta la función PVR y en consecuencia la aplicación ofrecería la opción de grabar, o si soporta la función de descarga, la aplicación al lado de cada vídeo podría poner un enlace para descargar el contenido. El acceso a estas funciones tiene que seguir siendo según el marco de seguridad visto anteriormente.

Asociación de APIs con protocolos

- Se soportan los siguientes tipos de URL: *http*, *https*, *dvb*. También se soporta *rtsp* en los terminales que tengan esta función incluida.

Extensiones y capacidades relativas al CEA-2014

- Posibilidad de indicar las capacidades del sintonizador DVB (*SD*, *HD*, etc...).
- Posibilidad de indicar las capacidades PVR (si están implementadas).
- Posibilidad de indicar las capacidades de descarga CoD (si implementadas).
- Calificación moral.
- API para metadatos.
- API para configuración.
- Posibilidad de indicar las capacidades DRM.
- Posibilidad de indicar la capacidad multimedia.
- Posibilidad de indicar otro tipo de capacidades.

Stream Events

- Es posible añadir/eliminar procesos que busquen la presencia de Stream Events en Transport Stream del flujo DVB.
- Se puede crear un Stream Event interno.
- Es posible recuperar datos de un carrusel usando un método XMLHttpRequest. Esto puede ser muy útil, ya que permite recibir y actualizar datos en una aplicación cuando no hay conexión a banda ancha. Podría servir para un ticker de noticias, información del tráfico y del tiempo. Además es muy inmediato, ya que suelen ser poca cantidad de datos que se reciben de forma inmediata en el stream DVB.

DRM

- Extensiones al objeto *application/oipfDRMAgent* para el pago de contenidos.

Todo lo anterior es un resumen de las funcionalidades más representativas que deben estar implementadas en el navegador HbbTV. Como hemos comprobado, varias son heredadas del estándar CEA-2014 pero la gran mayoría vienen adaptadas y modificadas de la especificación OIPF DAE. Lo que esto representa en la práctica, es lo que van a poder hacer las aplicaciones, a que partes del sistema pueden acceder, con que condiciones de acceso y como va a ser la interacción del usuario.

4.5.2.2 OIPF Media Formats

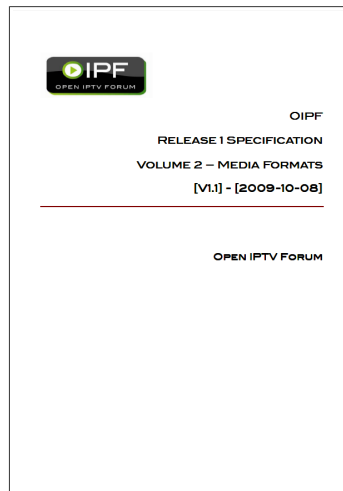


Figura 4.9 – Portada de la especificación OIPF Vol. 5 - DAE.

El estándar HbbTV incorpora de esta especificación los formatos para las esencias de video, audio y sus contenedores. Estos formatos solo se aplican al contenido On-demand (streaming, descarga, directos) del canal broadband. No especifica nada acerca de los formatos de audio/video para el canal broadcast, que se registrará por otras especificaciones. Aún así hay alguna influencia del canal broadcast en los formatos de audio y subtítulos para el canal broadband..

Los formatos de vídeo tienen en común que el codec de vídeo utilizado es el *H.264* (estándar ISO/IEC 14496-10 *MPEG-4* part 10), para el audio hay varios formatos. Y en cuanto a los formatos del contenedor, se encapsulará usando MPEG-2 Transport Stream o *MP4*.

4.5.2.2.1 Video

En el apartado de codec y propiedades del video, HbbTV adopta por completo lo definido en la especificación OIPF Media Formats, no hace ninguna modificación ni restricción.

Debido a esto, HbbTV incluye dos perfiles de video. Uno es un perfil para video SD y otro para video HD. Sus nombres (etiquetas) son AVC_SD_25, AVC_SD_30 para los perfiles SD y AVC_HD_25, AVC_HD_30 para los perfiles HD. Las versiones AVC_SD_30 y AVC_HD_30 son para sistemas a 30Hz (EEUU, Japón,...) aquí vamos a centrarnos en los Europeos AVC_SD_25 y AVC_HD_25, ya que el estándar HbbTV no las menciona explícitamente. Algo que ambos tienen en común son las restricciones a la estructura del GOP (Group of pictures).

En codificaciones MPEG como es H.264 (AVC) el proceso de codificación de video necesita de distintos tipos de fotogramas que permitan la codificación diferencial y bidireccional además de minimizar la propagación de errores. Estos distintos tipos de fotogramas se dividen en dos grupos: inter-frame e intra-frame.

Un fotograma intra-frame (normalmente llamados I) es independiente de los fotogramas que le preceden y los que le suceden, no depende de la información de ningún otro fotograma.

Los fotogramas inter-frame si que dependen de la información de los fotogramas que les preceden y/o de los que les suceden. En H.264 los hay de dos tipos B y P. Los B son de predicción bidireccional y pueden ser interpolados con la información de los fotogramas precedentes y sucesores. Los inter-frame P (Predicted pictures) son fotogramas interpolados por la información de fotogramas I o P previos.

La cantidad de datos necesaria para la interpolación de los fotogramas B y P es el resultado de aplicar coeficientes de transformación y vectores de movimiento que corrigen el movimiento entre fotogramas y que implica el uso de técnicas de compensación de movimiento.

Dicho esto, hay dos tipos de *GOP*: el *GOP* cerrado y el *GOP* abierto. El *GOP* cerrado es un grupo cerrado de fotogramas dentro del cual los inter-frame y los intra-frame tienen un orden concreto y no necesitan de información de fuera del grupo. Un *GOP* abierto no tiene esta restricción y es más eficiente a la hora de predecir fotogramas pero en transmisión hace que sea más difícil particionar el flujo de video.

Un flujo de video codificado es una secuencia de *GOP*s. El tamaño del *GOP* suele ser de entre 12 y 15 fotogramas. Un ejemplo:

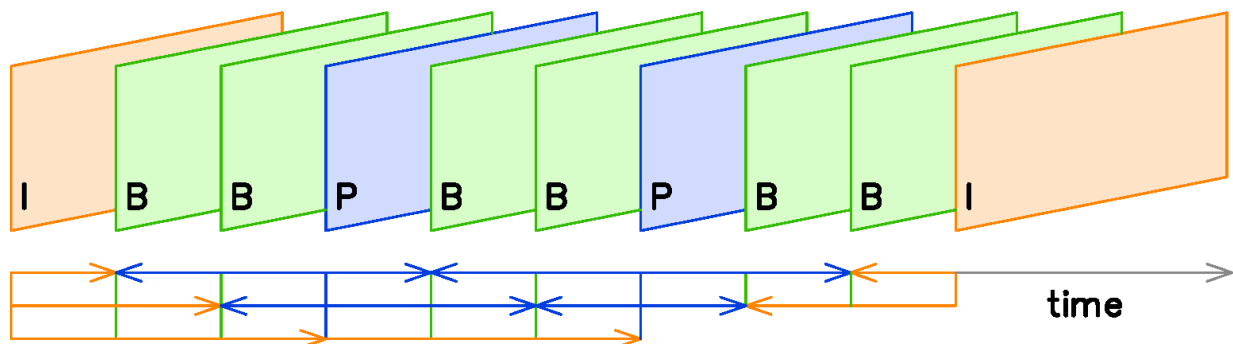


Figura 4.10 – Estructura de un *GOP*.
http://es.wikipedia.org/wiki/Group_of_pictures

En este *GOP* las flechas inferiores indican las referencias predictivas que hay entre ellas. Se ha escogido una ordenación *IBBPBBPBB* a modo de ejemplo, ya que permite establecer lazos de predicción más usuales.

Las características del *GOP* pueden representarse por una pareja $M=3, N=9$. El primero de ellos nos dice la distancia que hay entre dos imágenes tipo I o P. El segundo nos dice la distancia que hay entre dos imágenes enteras, es decir, entre dos imágenes tipo I y es la que nos dirá la longitud del *GOP*. La elección del *GOP* tiene mucho que ver con el trickplay que se vaya a hacer del stream de video.



El OIPF Media Formats que HbbTV ha adoptado para dar formatos a los contenidos A/V del canal broadband define las siguientes restricciones en la estructura del GOP, que son comunes a ambos perfiles SD y HD:

- Los fotogramas I tendrán serán identificados con `slice_type=7`.
- Los fotogramas P tendrán serán identificados con `slice_type=5`.
- Los fotogramas A serán identificados con `slice_type=6`.
- La decodificación de fotogramas I o P se hará según su orden de presentación.
- Los fotogramas P no derivarán de los B.
- La pareja de campos de referencia complementaria que incluyan campos I/P no incluirán campos B (para entrelazados).
- Los fotogramas B (completos) derivarán de fotogramas I/P o de parejas de campos de referencia de la pareja de campos de referencia complementarios.
- El campo de referencia B (entrelazado) puede derivar del campo de referencia de la pareja de campos de referencia complementarios.

Perfil SD H.264/AVC (AVC_SD_25)

Es un perfil de resolución SD cuyas características son las siguientes:

- Perfil y nivel: Main 3.0 y High cuando lo permita el receptor.

El perfil se refiere al conjunto de características usadas para la codificación en ese formato. En el estándar H.264 hay 21 tipos distintos de perfiles, que definen p.ej el formato de color (4:2:0, 4:2:2, 4:4:4), resolución en bits de la muestra (8, 10, 12), tipo de entrelazado, si se hace una codificación predictiva sin pérdidas, etc. Perfiles típicos de entornos no profesionales son Baseline, Main y High.

Baseline: Se suele usar en dispositivos con pocos recursos como móviles o en aplicaciones de videoconferencia.

Main: Este perfil era el más utilizado para aplicaciones de broadcast doméstico y en formatos de almacenamiento. Pero ha sido reemplazado por el perfil High.

High: Actualmente es el perfil más usado en broadcast y almacenamiento, sobre todo en para HD. Este es el perfil usado por DVD HD (totalmente en desuso) y Blu-Ray.

El nivel se refiere al conjunto de requisitos que se le va a exigir a un decodificador para que sea capaz de decodificar la señal correctamente. Al asignar un nivel a un perfil, lo que estás definiendo es la resolución máxima, la tasa de cuadros, la tasa de bits del flujo, que debe ser capaz de ofrecer el decodificador para poder decodificar la señal correctamente. Características definidas en los niveles son por ejemplo, máxima cantidad de macrobloques, máxima cantidad de muestras de luma (componente de color de la señal), máxima tasa de bits del flujo. De esto salen definiciones del tipo 720x576@25 (máxima_resolución@máxima tasa de fotogramas).



- Relación de aspecto: 4:3 o 16:9.
- Tasa de fotogramas: 25Hz (25fps) Progresivo o entrelazado (50i).
- Resoluciones admitidas (HxV): 720x576, 544x576, 480x576, 352x576 y 352x288

Perfil HD H.264/AVC (AVC_HD_25)

Es el perfil de resolución HD cuyas características son las siguientes:

- Perfil y nivel: High 4.0
- Relación de aspecto: 16:9
- Tasa de fotogramas: 25Hz Progresivo o entrelazado (50i) y 50Hz progresivo (50p).
- Resoluciones admitidas: (HxV) 1920x1080, 1440x1080, 1280x1080, 960x1080, 1280x720, 960x720, 640x720 (todos 16:9)

4.5.2.2.2 Audio

Se soportan cuatro codificaciones distintas para el audio: *HE-AAC/AAC*, *AC-3*, *MP2*, *MP3* y *PCM*. Pero HbbTV no hace referencia explícita a todas.

HE-AAC y AAC

Es una codificación de audio basada en ISO/IEC 14496-3:2005. Ofrece dos objetos de audio: AAC LC (Advanced Audio Coding Low Complexity) y AAC SBR (AAC Spectral Band Replication). De las dos codificaciones de audio el perfil AAC es el perfil básico que luego ha sido mejorado y renombrado a HE AAC (High Efficiency AAC). El perfil AAC aporta el objeto AAC LC y el perfil HE-AAC aporta el perfil AAC SBR.

HE-AAC está considerado por la industria como un codificador de audio de alta calidad. Las codificaciones stereo entre 16 y 48 Kbps se consideran aceptables. Las frecuencias de muestreo van desde 24 a 96 Khz. Permite las codificación de canales Mono, Stereo y Multicanal (5.1 y 7.1).

Hay otro perfil más, el HE-AAC v2, pero no se puede usar en HbbTV, respetando así una de las premisas para la implementación del sistema, que es la complejidad mínima de implementación del terminal.

La etiqueta utilizada en HbbTV para identificar este audio es HEAAC.

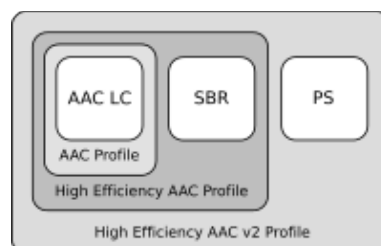


Figura 4.11 – Relación entre las versiones de codec AAC.
<http://es.wikipedia.org/wiki/HE-AAC>



Para el contenido A/V el audio HEAAC utilizará la codificación de Nivel 4 definida en el estándar ISO/IEC 14496-3:2005. Para los clips de audio utilizará el Nivel 2 de la misma especificación.

AC-3

Este formato de sonido está basado en el estándar ATSC A/52B. En el Anexo E de este estándar también se define el Enhanced AC-3 (E-AC-3).

Es un algoritmo de compresión digital que puede codificar desde uno hasta 5 canal de audio con ancho de banda de frecuencias completo junto con un canal mejorado de bajas frecuencias. Estos seis canales pueden codificarse en PCM o en un flujo de bits con tasas de 32 Kbps hasta 640 Kbps. Cuando el flujo o la codificación incluye los 6 canales de audio se conoce comúnmente como 5.1.

Fue desarrollado por los laboratorios Dolby y es conocido comercialmente como Dolby Digital. El ATSC no aceptó este nombre al incluirlo en el estándar y por ello el estándar se llama ATSC A/52.

HbbTV además da soporte a E-AC-3 a diferencia del OIPF que no lo soporta. El E-AC-3 es una mejora del AC-3 que proporciona mejor calidad a igual tasa de bits además de poder codificar hasta 8 canales (escalable a 16) de audio de hasta 6 Mbps. El E-AC-3 también está estandarizado por la ETSI en ETSI TS 102 366 v1.2.1.

Este será el formato de preferencia en el caso de ofrecer contenidos multicanal por el canal broadband.

Para este caso, HbbTV impone una serie de condiciones:

- Si el terminal tiene una salida stereo, deberá hacer downmix del contenido multicanal a stereo (5.1 → 2.0)
- Si el terminal dispone de una salida de audio digital deberá poder hacer pass-through (dejar pasar la señal tal cual) del flujo de audio multicanal a la salida digital. Además si el contenido multicanal es en formato HE-AAC deberá transcodificarlo a formato AC3, que es el que entienden los equipos externos tipo Home Cinema.
- El terminal deber ser capaz de controlar el downmix con los metadatos que incluya el flujo para esta finalidad. También debe ser capaz de hacer pass-through de los metadatos cuando sea necesario.

La etiqueta de este formato en el sistema es AC3.

MPEG-1 Audio Layer II (MP2)

Es un formato de compresión de audio con pérdida definido en el estándar ISO/IEC 11172-3:1993/Cor 1:1996, "Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio".

Puede codificar canales Mono, Joint Stereo, Stereo o Multicanal.

En cuanto al *bitrate* el MPEG-1 Layer II puede usar tasas de bits desde 32 Kbps hasta 384 Kbps con frecuencias de muestreo de 32, 44,1 o 48 Khz para el canal principal y opcionalmente 16, 22,05, 24, 32, 44,1 o 48 Khz para el canales secundarios.



No hay que confundirlo con el formato MPEG-2 (ISO/IEC 13818) usado habitualmente en los TS de las emisiones de televisión SD DVB-T/C/S y en los PS (Program Streams) de DVDs. MPEG-2 define formatos de audio (ISO/IEC 13818-3: MPEG2 Part 3 e ISO/IEC13818-7: MPEG2 Part 7), vídeo (ISO/IEC 13818-2: H.262/MPEG2 Part 2) y sus contenedores (ISO/IEC 13818-1: TS y PS).

La etiqueta de este formato en el sistema es MPEG1_L2.

MPEG-1 Audio Layer III (MP3)

En HbbTV el MP3 (ISO/IEC 11172-3) únicamente puede ser utilizado en servicios exclusivos de audio, sin vídeo. No está permitido utilizarlo junto con un flujo de vídeo para formar un servicio A/V. El MP3 es un formato de uso muy habitual en entornos no profesionales donde permite la codificación de audio a tasas de bits muy reducidas con un buen resultado en la percepción de audio subjetiva. Ofrece la posibilidad de codificar el contenido de un CD, con una percepción de calidad muy buena ocupando 11 veces menos. Puede codificar audio en bitrates que van desde 32 a 320 kbps con frecuencias de muestreo de 32, 44,1 y 48 Khz. El formato más habitual es 128 Kbps / 44,1 Khz.

Desbancó en su momento al MP2 y hoy día está empezando a ser desbancado por AAC por ser más eficiente y tener mejor calidad a bitrates más bajos.

La etiqueta de este formato en el sistema es MPEG1_L3.

WAVE (WAV)

El formato WAV, PCM sin comprimir puede utilizarse para notificaciones de audio dentro del DAE (embebiendo un clip dentro del código de la aplicación). En caso de usarse se hará con el siguiente formato:

- Frecuencia de muestreo: 12 Khz.
- Codec: PCM sin comprimir o ADPCM.
- Bits de cuantificación: 16 bits.
- Canales: 1 (Mono).

La etiqueta de este formato en el sistema es WAV.

4.5.2.2.3 Contenedores

Un contenedor permite el transporte de contenido A/V e información relacionada (subtítulos, *PMT*, *EPG*, ...) de una forma estructurada. HbbTV incluye para el canal broadband dos formatos de contenedores.

Los dos contenedores válidos para aplicaciones HbbTV son el MPEG-2 Transport Stream (conocidos como *M2TS* o TS) y el MP4.



MPEG-2 Transport Stream

El MPEG-2 Transport Stream (TS de ahora en adelante), es un formato estandarizado para la transmisión y el almacenamiento de flujos de video, audio y datos. Es ampliamente utilizado en los distintos sistemas de radiodifusión existentes DVB, ATSC e IPTV. Está especificado en el ISO/IEC 13818-1 (MPEG-2 Part 1) y también es conocido como H.222.0 (por la ITU).

Un TS lleva dentro varios flujos que pueden ser de video, audio y datos. Cada uno de estos flujos se divide en paquetes más pequeños y estos paquetes a su vez se vuelven a dividir en otros paquetes más pequeños para ser transmitidos con control de errores.

Hay unos datos obligatorios que tienen que ir dentro del TS y que sirven para identificar todos los flujos (*PID*) y la estructura del propio TS. Son los denominados *PSI* (Program Specific Information) y dicen que es cada flujo y que flujo va asociado a cada servicio.

HbbTV define el uso del TS solo para el canal broadband, con las siguientes restricciones:

- El TS solo puede contener un PMT. El PMT es un tipo de tabla PSI que dice que flujos de audio/video/datos corresponden con un programa en concreto.
- Se debe aplicar el perfil "TS Optional-SI" definido en el ETSI TS 102 304. Esto significa que el TS debe incluir un PAT y un PMT. El PAT es un índice de PMTs. También dice que el DVB-SI (ETSI EN 300468) es opcional y que el EIT (EPG) es opcional.
- La tasa de datos máxima para un TS con contenido SD o HD no puede superar los 8 Mbps. Esto es incluyendo overheads del protocolo TCP, HTTPS, etc...
- La tabla AIT (lo veremos en el punto 4.5.3) que esté dentro del TS no tendrá que ser procesada por el terminal cuando el TS se reciba por el canal broadband.
- No es un requisito que el terminal soporte la extensión "time stamped MPEG-2 TS"
- Es posible la encriptación (DRM) de contenidos.

Formato de archivo MP4

El MPEG-4 Part 14 (MP4) especificado en el estándar ISO/IEC 14496-14:2003 es un contenedor multimedia estandarizado para el almacenamiento de flujos de audio y video digital, especialmente para los codificados siguiendo los estándares MPEG aunque es válido para otros flujos distintos como flujos de datos de subtítulos o imágenes.

Es un tipo de contenedor que favorece el streaming de sus contenidos con una pista adicional (*hint track*) que contiene la información necesaria para que el archivo pueda ser transmitido en forma de streaming.

El MP4 del que estamos hablando aquí es una extensión del ISO Base Media File Format (*ISO BMFF*, ISO/IEC 14496-12:2004), que define una estructura de contenedor más general. El ISO BMFF a su vez fue creado como una generalización del MP4 definido en el ISO/IEC 14496-1:2001 (MPEG-4 Part 1: Systems) que estaba totalmente basado en el formato QuickTime de Apple. Se podría decir que el MP4 que tratamos aquí es un MP4 v2 basado en el ISO BMFF que está basado en MP4 v1 que fue creado partiendo del formato QuickTime.

En la siguiente figura podemos ver la relación entre ISO, MP4, AVC, QuickTime y MPEG-21:

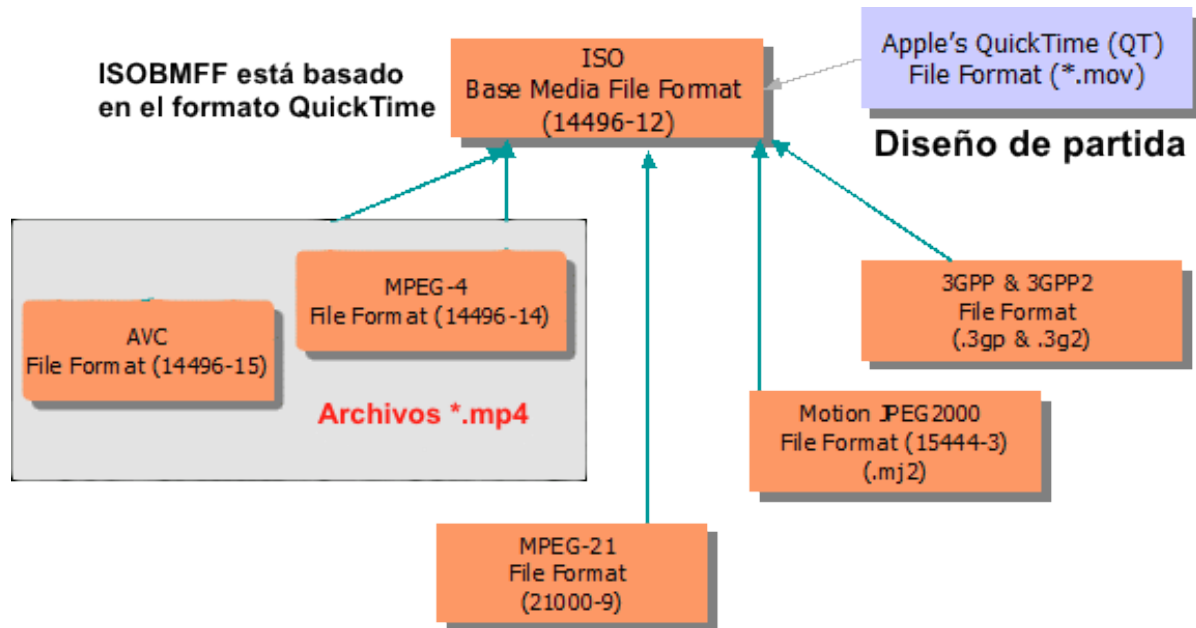


Figura 4.11 – Relación entre ISO, MP4, AVC.

<http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-ff/index.htm>

Un archivo mp4 está organizado internamente en objetos. La estructura de esos objetos varía en función del tipo de objeto que sea. Los objetos son denominados “**box**” en el caso del ISO BMFF/MP4 y “**atom**” en el caso de los archivos QuickTime. Todos los datos del archivo están dentro de algún “box”, no hay nada fuera. Cada “box” está identificado según su función. Se pueden organizar jerárquicamente de tal forma que haya “box” que son “hijos” de otros “box”. Algunos de los “box” más representativos:

- *ftyp* (File Type Box): Se pone al principio del archivo para identificar es tipo de codificación usada, la compatibilidad o el uso pretendido del archivo.
- *free*: Espacio no usado y disponible.
- *ctab*: Tabla de color de la pista.
- *mdat*: Contenedor de datos A/V.
- *meta*: Contenedor de metadatos.
- *moov*: Contenedor de todos los metadatos.
- *pdin*: información para la descarga progresiva (streaming)
- *sdvp*: Perfil SD
- *stco*: información parcial de offset de datos
- *moof*: Fragmento de película

Hay una lista completa de todos los tipo de box en:

<http://www.mp4ra.org/atoms.html#qtatom>.



HbbTV impone algunas condiciones de uso al formato MP4:

- Se debe usar el "box" *stco*, y no se puede usar el *co64*.
- Para permitir la reproducción de contenidos por descarga progresiva (reproducir sin necesidad de haber descargado el archivo por completo) se deben usar los "box" *moov* y *moof* y se recomienda el uso del "box" *pdin*.
- También se proveen métodos de protección (DRM, con base en el DRM de Marlin).
- En caso de contener E-AC-3 deberá ser acorde a TS 102 366.
- El tamaño del "box" *moov* no debería ser mayor de 2,5 MB, ya que ralentizaría el inicio de la reproducción si la conexión a banda ancha no es muy rápida.
- Se puede utilizar la propiedad *largesize* del estándar ISO BMFF. El tamaño máximo de un "box" no debería pasar de 4 GB. (OIPF dice justo lo contrario)

4.5.2.2.4 Subtítulos

En la versión 1.0 del estándar HbbTV no se especifica nada acerca de los subtítulos, pero en la errata si. Lo que aporta la errata es que solo se soportan los subtítulos para contenidos recibidos por el canal de banda ancha en formato TS. Para el formato MP4 no se define nada. En cuanto al formato de subtítulos, el terminal deberá soportar los mismos formatos que soporta para los subtítulos del canal broadcast.

La realidad técnica actual es que no hay un formato estandarizado para los subtítulos en el canal de banda ancha. Hay un tipo de subtítulos denominado TTML pero no está estandarizado y los fabricantes no lo tienen desarrollado a nivel técnico como para poder ser usados en HbbTV. Es decir, puede que las soluciones de accesibilidad para subtítulo de los contenidos on demand estén disponibles pasado un tiempo de que HbbTV esté ya disponible en el mercado.

4.5.2.2.5 Resumen de formatos soportados en HbbTV

En las siguientes dos tablas se definen las combinaciones de formatos de video/audio/contenedores y subtítulos que los receptores HbbTV deben soportar como mínimo para el canal broadband:

Contenedor	Formato de video	Formato de audio	Formato de subtítulos	Tipo MIME
TS	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (si está soportado en el canal broadcast)	El mismo que el soportado para el canal broadcast.	video/mpeg
MP4	AVC_SD_25 AVC_HD_25	HEAAC E-AC3 (si está soportado en el canal broadcast)	No se define nada.	video/mp4

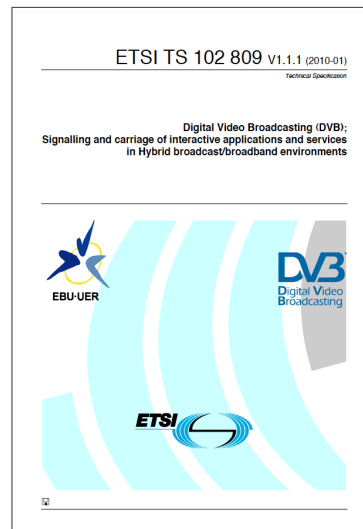
Tabla 4.1 – Combinaciones de formatos definidas por HbbTV.

y para el caso de servicios de solo audio y clips de audio:

Formato de audio	Tipo MIME
MPEG1_L3	audio/mpeg
HEAAC	audio/mp4

Tabla 4.2 – Combinación de formatos para servicios de audio exclusivos definidos por HbbTV.

4.5.3 ETSI TS 102 809 – Señalización de aplicaciones



Todo lo que hemos visto hasta ahora son funciones relativas al navegador. HbbTV necesita también integración con el entorno DVB. Para ello, se incorporan las funciones del estándar *ETSI TS 102 809 “Signalling and carriage of interactive applications and services in HBB environments”*. Este estándar fue creado por el DVB en Marzo de 2009 y publicado por la ETSI como estándar en Febrero de 2010.

Este estándar define como tienen que señalizarse las aplicaciones que se ejecutan en el contexto de un programa de TV/radio que viaja en su correspondiente múltiplex DVB. Al igual que se hacía en MHP, la señalización de aplicaciones interactivas se hace usando una tabla creada para tal efecto: la AIT (Application Information Table) que va incluida en el servicio DVB y que está listada en el PMT (Program Map Table).

La AIT incluye la señalización de todas las aplicaciones que se quieran ejecutar en el contexto de ese programa. Dicho de otro modo, las aplicaciones solo se ejecutarán si se encuentran señalizadas en la AIT. Gracias a esto, se puede evitar que una aplicación de terceros “secuestre” la señal de programa. Como hemos dicho antes, el estándar OIPF permite que desde una aplicación interactiva ejecutándose en el contexto de un programa A se pueda sintonizar otro programa B. Pero si en la AIT del programa B no está señalizada la aplicación, ésta será detenida y no podrá llevar a cabo el cambio de programa. Con esto se pretende que no haya overlays no autorizados, preservando la integridad de la señal. Una de las aplicaciones señalizadas en la AIT puede ser marcada como “autostart”. Esto significa que la aplicación se ejecutará automáticamente al sintonizar el servicio siempre y cuando en la configuración del terminal no se haya determinado lo contrario, es decir, no iniciar automáticamente las aplicaciones “autostart”.

Respecto a las aplicaciones “autostart”, se ha llegado a un acuerdo por el cual lo único que hacen es dibujar un pequeño icono en el que se muestra un botón rojo (denominado “gancho” o “hook”) invitando al usuario a usar el servicio y que desaparece tras unos segundos. Esto se hace para no molestar a los usuarios con sobreimpresiones no deseadas y por homogeneizar la presentación de la aplicación “autostart”. Tras la desaparición del icono con el botón rojo, el usuario puede seguir arrancando la aplicación en cualquier momento ya que al aplicación está ejecutándose en modo transparente pero reaccionará al *keyevent* de pulsar el botón rojo en cuanto este se produzca.



El AIT también incorpora una opción por la cual una aplicación HbbTV puede ser marcada como “digital teletext application”, de esta forma una aplicación HbbTV puede sustituir al teletexto tradicional. El comportamiento definido para el caso en que haya presente una aplicación de teletexto HbbTV y el teletexto tradicional es que tras una primera pulsación del botón “TXT” (o similar) del mando se iniciará el teletexto HbbTV, tras una segunda pulsación se presentará el teletexto tradicional. Si no hay teletexto HbbTV se iniciará directamente el teletexto tradicional.

Además de la señalización de aplicaciones, el ETSI TS 1102 809 especifica el transporte de aplicaciones mediante el canal broadcast DVB, siendo esta una opción interesante para los terminales que no tengan conexión a banda ancha. Al igual que en MHP, para transmitir las aplicaciones se usa el carrusel de objetos del protocolo DSM-CC. La cantidad de datos que es posible transmitir por este medio es claramente inferior a lo que permite el canal broadband pero aún así es suficiente para transmitir aplicaciones con poco peso como pueda ser el teletexto digital, EPGs o la aplicación de cita médica de 7RM desarrollada para MHP.

Una parte del sistema DSM-CC son los “stream events” que ya hemos mencionado varias veces a lo largo de este texto. Los stream events son pequeños paquete de datos que se pueden transmitir sincronizados con la señal de programa. Permiten que la aplicación interactiva muestre informaciones o datos en un momento preciso sin retardo. Es por ello que a los stream events también se les conoce como “Do it now” events: eventos de “Hazlo ya”. La capacidad de enviar eventos de forma inmediata y síncrona con lo que está pasando en el programa da lugar a aplicaciones tipo televoto, o encuesta. El canal broadband no es válido para este tipo de aplicaciones porque tiene retardos variables, además de que necesitaría que los servidores web estuviesen dimensionados de tal forma que fuesen capaces de enviar información a millones de conexiones de forma simultánea. El uso del canal broadcast es mucho más eficiente.

Por tanto este estándar aporta lo siguiente a HbbTV:

- Señalización de aplicaciones en el flujo DVB del canal broadcast (AIT).
- Capacidad para iniciar y detener las aplicaciones en el contexto de un programa de TV/radio (usando la AIT).
- Señalización de Teletexto Digital HbbTV (en la AIT).
- Transmisión de los datos de las aplicaciones en el canal DVB (mediante DSM-CC).
- Transmisión de stream events síncronos con la señal de TV/radio (con DSM-CC).

Vamos a analizar un poco más en profundidad estos tres conceptos: AIT, DSM-CC y stream events. Pero previamente vamos a dibujar el escenario sobre el que nos estamos moviendo, el TS del canal broadcast.

Cada flujo elemental (Elementary Stream) de video, audio o datos que viaja dentro del TS tiene asociado un identificador único (PID, 13 bits). El TS tiene que multiplexar los flujos elementales de audio, video y datos. Para hacer el proceso de multiplexación más eficiente, se divide cada flujo en paquetes dando lugar a unos flujos elementales divididos en paquetes (*PES*, packetized elementary stream).

Para poder transmitir esos PES, hay que volver a dividirlos en paquetes de un tamaño óptimo para la transmisión. El tamaño del paquete para transporte es de 188 bytes, que es bastante más pequeño que el tamaño de un PES.

Por ello, cada PES se transmite utilizando varios paquetes de transporte. Cada uno de esos paquetes de 188 bytes empieza siempre con un byte de sincronismo (0x47). El TS provee de mecanismos de corrección de errores y sincronización para evitar la degradación de la señal.



El TS proporciona una representación lógica del contenido que transporta, girando entorno al concepto de programas/servicios. Para ello utiliza la tabla PMT (Program Map Table) que lista todos los PIDs que forman un programa. El PMT también tiene su propio PID. Por ejemplo, un TS usado en televisión digital puede transportar 3 programas que representarían 3 canales de televisión. Cada uno de esos canales está compuesto por un flujo de vídeo, uno o dos flujos de audio y otros flujos con metadatos. Para que el terminal pueda presentar cualquiera de esos 3 canales mira el PMT para saber exactamente que PIDs tiene que decodificar y cuales tiene que descartar. Y después tenemos la PAT (Program Association Table) que es un índice de todos los servicios/programas que se encuentran en el TS. El siguiente gráfico lo ilustra todo:

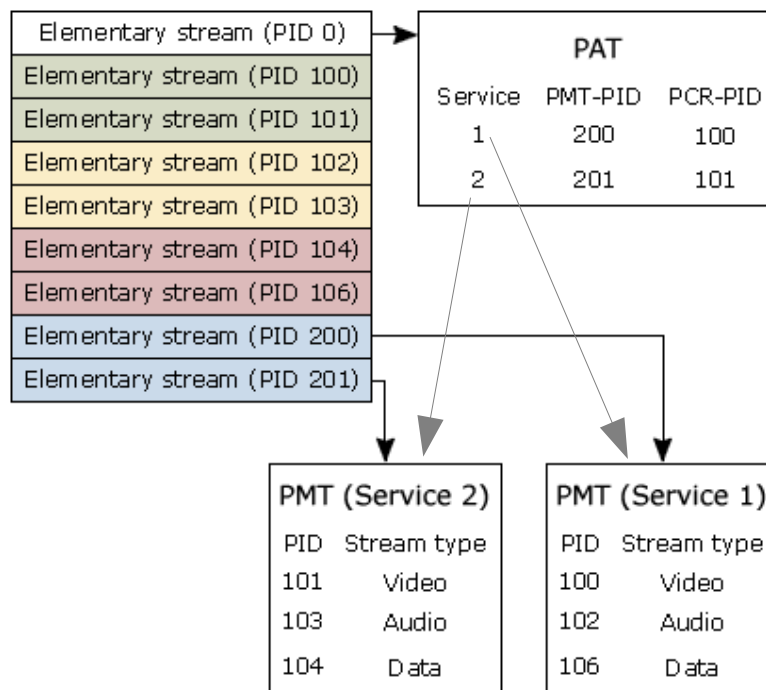


Figura 4.12 – Representación de los distintos TS y su organización en PMT y PAT.
http://www.mhp-interactive.org/tutorials/dtv_intro/dtv_intro

En conjunto todos los tipos de Tabla que acabamos de mencionar reciben el nombre de “SI table” (Service Information Table). Hay más además de las mencionadas:

- PAT: Program Association Table (siempre con PID 0).
- PMT: Program Map Table.
- CAT: Conditional Access Table
- NIT: Network Information Table.
- SDT: Service Description Table.
- EIT: Event Information Table (EPG)
- BAT: Bouquet Association Table
- TDT: Time and Date Table
- TOT: Time Offset Table

Las tablas PAT + PMT + CAT son conocidas como PSI (Program Specific Information).



4.5.3.1 Señalización AIT

Antes de que el gestor de aplicaciones del navegador HbbTV pueda ejecutar cualquier aplicación, el receptor debe saber que tal aplicación existe. Después debe saber si el usuario puede ejecutarla o no es ese momento y finalmente debe saber si dispone de todo lo necesario para poder ejecutarla. Las dos primeras premisas y parte de la tercera se controlan con una tabla SI extra creada para este propósito, la AIT (Application Information Table). Esta tabla contiene una entrada por aplicación y se difunde con cada servicio que contenga una aplicación interactiva. Va listada en la tabla PMT del servicio correspondiente. La AIT contendrá toda la información necesaria para que el receptor pueda ejecutar la aplicación y que el usuario sepa que aplicaciones tiene disponibles en cada momento. Entre esta información se cuenta el nombre de la aplicación, la ubicación de sus archivos y cualquier dato que deba ser pasado a la aplicación al arrancarla.

Cada aplicación tiene asignado un identificador único que es almacenado en la AIT. Esto permite que otras partes del sistema puedan acceder a la aplicación de forma unívoca. Este identificador se compone de dos campos un campo de 32 bits para identificar la organización que realiza la aplicación y que debe ser único y un campo de 16 bits que identifica la aplicación. El campo de 16 bits no tiene por que ser único pero en caso de que la misma AIT señalice más de una aplicación no deben coincidir. El terminal puede arrancar o parar automáticamente las aplicaciones en función del código de control señalado en la AIT. Los códigos de control dicen cuando una aplicación puede arrancarse automáticamente (“AUTOSTART”), cuando debe finalizar si está en ejecución y cuando puede ser iniciada manualmente. Así el radiodifusor puede, por ejemplo, activar una aplicación durante un intervalo de tiempo determinado.

A continuación listamos los descriptores y entidades de señalización cuya codificación MPEG-2 debe ser soportada por HbbTV:

- Application type: *application_type* sirve para saber si el terminal es compatible con el tipo de aplicación. En el caso de HbbTV se señalará con el código *0x0010*.
- Application ID: Aquí es donde se identificad a la organización responsable de la aplicación (*organisation_id*) y a la aplicación (*application_id*). Para el *application_id* hay varios código que indican si la aplicación está firmada o si no lo está. En HbbTV las aplicaciones firmadas podrás ser ejecutadas como de confianza. Las aplicaciones marcadas como no firmadas podrás ser ejecutadas pero no como aplicaciones de confianza (no tendrán acceso a recursos privilegiados del terminal) Las aplicaciones que estén marcadas como algo que no sea firmado o no firmado, a no ser que sean requeridas por otra especificación, no deberán iniciarse y serán descartadas por el terminal (errata HbbTV).
- Código de control de las aplicaciones:
 - 0x01: AUTOSTART → Si el terminal lo permite se iniciará sola.
 - 0x02: PRESENT → Está presente pero no se ejecuta sola.
 - 0x04: KILL → En cuanto aparezca este código en la AIT la aplicación morirá.
 - 0x07: DISABLED → La aplicación no se puede ejecutar.

Del estándar DVB se caen los estados DESTROY, PREFETCH, REMOTE y PLAYBACK_AUTOSTART. El flujo entre estos estados viene definido por el ciclo de vida de la aplicación.



- Perfiles de la plataforma:
 - 0x0000: Para aplicaciones que no necesiten más que el perfil básico..
 - 0x0001: La aplicación requiere la funcionalidad de Descargas.
 - 0x0002: La aplicación requiere la funcionalidad PVR.
 - 0x0004: La aplicación requiere la funcionalidad RTSP.
- Visibilidad de la aplicación: Solo el válido el código *VISIBLE_ALL*. Aplicación visible a los usuarios y a otras aplicaciones mediante un API.
- Prioridad de la aplicación: Que aplicación se inicia primero, en caso de quedarse sin recursos cual se cierra primero, cual se inicia si tienen el mismo id....
- Uso de la aplicación: Código para identificar si es un servicio conocido como Teletexto, EPG o Chat. Se usará el código *0x01* → Teletexto.
- Program Specific Information: El PMT de un servicio DVB que incluya una o más aplicaciones interactivas deberá incluir la localización del TS que lleve la tabla AIT y la ubicación de los TS que lleven los datos de la aplicación.
- Application Information Table: La AIT está compuesta por distintas subtablas que llevan la información de los datos de la aplicación, de los códigos de estado, etc... Para el *application_type* se transmitirá un máximo de una subtabla (se usará un solo PID) HbbTV por servicio. Todas las secciones de las subtablas AIT HbbTV se transmitirán 1 vez por segundo.
- Application Signalling Descriptor: El *application_signalling_descriptor* se usa dentro del loop del flujo elemental de la PMT donde el *stream_type* del flujo elemental es 0x05 indicando así que el flujo elemental es un AIT.
- Application Descriptor: Debe haber al menos un *application_descriptor* dentro de cada loop de descripción de la AIT.
- Application usage descriptor: Esto sirve para el caso en el que el mando a distancia no tiene ningún botón específico para acceder a servicios de los denominados “well known” como el Teletexto o el EPG. En este caso se da la opción de usar este identificador para que desde la aplicación pueda lanzarse dicho servicio. Se utilizará únicamente el código *0x01* como antes para identificar el servicio de Teletexto.
- User information descriptor: Se pueden usar para mostrar información que se considere útil para el usuario.
- Transport Protocol Descriptors: Se usarán los siguientes:
 - 0x0001*: Carrusel de objetos sobre el canal broadcast.
 - 0x0003*: HTTP sobre el canal broadband.



- Simple application location descriptor: El bucle AIT deberá contener siempre uno. Consta de dos etiquetas → *descriptor_tag*: 15 bits con valor *0x15* e *initial_path_bytes*: Que contienen una cadena especificando la URL del documento raíz de entrada. **Este es el descriptor HbbTV que se usará para lanzar aplicaciones por el canal broadband.**
- Simple application boundary descriptor: Solo de admiten los prefijos “*dvb://*”, “*http://*” o “*https://*”. Y solo se soportarán prefijos que hagan referencia a dominios de segundo nivel, se ignorarán los elementos en la raíz.

También es posible señalar aplicaciones por el canal broadband usando la llamada codificación XML del AIT. Esto significa que se podría leer un archivo XML del canal broadband para señalar artificialmente una aplicación. Esto se puede usar para señalar aplicaciones bc-independent o para señalar aplicaciones interactivas en entornos de televisión analógica, algo que se cubre en el Anexo C del estándar HbbTV. Señalar aplicaciones interactivas en entornos de televisión analógica tiene aplicación en mercados donde aún no ha llegado la televisión digital y se prevé que no vaya a hacerlo a corto plazo. A continuación pegamos un trozo de código en *Python* donde muestra como se señala una aplicación HbbTV en el *Opencaster* de Avalpa:

```
# parameters reported into ths AIT to signalize a broadband application.
appli_name = "application Portail" #application name; appli_root = "http://192.168.1.1/portail/" #URL base of
transport_protocol_descriptor; appli_path = "index.html" #initial_path_bytes of simple application descriptor; organisationId_1 = 10 # this
is a demo value. dvb.org should assign an unique value; applicationId_1 = 1001 # this is a demo value. This number corresponds to a
trusted application; applicationId_2 = 1002;
ait = application_information_section(
    application_type = 0x0010,
    common_descriptor_loop = [
        external_application_authorisation_descriptor(
            application_identifiers = [[organisationId_1,applicationId_1],
[organisationId_1,applicationId_2]],
            application_priority = [5, 1]),
        application_loop = [
            application_loop_item(
                organisation_id = organisationId_1,
                application_id = applicationId_1,
                application_control_code = 1, #AUTOSTART
                application_descriptors_loop = [
                    transport_protocol_descriptor(
                        protocol_id = 0x0003,
                        URL_base = appli_root,
                        URL_extensions = [],
                        transport_protocol_label = 3),
                    application_descriptor(
                        application_profile = 0x0000,
                        #0x0000 basic profile
                        #0x0001 download feature
                        #0x0002 PVR feature
                        #0x0004 RTSP feature
                        version_major = 1, # corresponding to version 1.1.1
                        version_minor = 1,
                        version_micro = 1,
                        service_bound_flag = 1, # the application is expected to die on service change
                        visibility = 3, # the applications is visible to the user
                        application_priority = 1,
                        transport_protocol_labels = [3], # If more than one protocol is signalled then each
protocol is an alternative delivery mechanism. The ordering indicates the
broadcaster's view of which transport connection will provide the best user experience
(first is best)),
                    application_name_descriptor(
                        application_name = appli_name,
                        ISO_639_language_code = "fra"),
                    simple_application_location_descriptor(initial_path_bytes = appli_path), ] ),
```



4.5.3.2 DSM-CC

DSM-CC (Digital Storage Media – Command and Control) en su origen fue diseñado para el control de VTRs (Video Tape Recorder) conectados en red. Desde su creación ha sido extendido y ahora es capaz de controlar servidores de video MPEG en red (reproducir, parar y pausar). También soporta la transmisión de datos usando MPEG-2, códigos de tiempos para video MPEG-2, broadcast de datos y de sistemas de archivos. Estas dos últimas funciones son las que nos interesan en HbbTV.

Para transmitir los datos en el canal broadcast el radiodifusor transmite periódicamente cada archivo del sistema de archivos y el receptor espera a que aparezca el archivo que necesita. Son las operaciones llevadas al cabo en el reproductor las que determinan que archivos son necesarios. A esta forma de enviar y recibir datos se le denomina carrusel. En DSM-CC los datos a transmitir se dividen en módulos, a cada módulo se le añade información y luego se transmiten siguiendo un orden.

DSM-CC soporta dos tipos de carrusel el carrusel de datos y el carrusel de objetos. El carrusel del datos es más sencillo, permite al radiodifusor enviar bloques de datos al receptor pero lo hace sin ningún tipo de indicación acerca del tipo de datos que son y es el receptor el que debe darle sentido a lo que ha recibido. Los carruseles de objetos son bastante más complejos y por tanto ofrecen una solución a escenarios más complejos. El carrusel de objetos se construye sobre el carrusel de datos y ofrece prácticamente las mismas funciones que un sistema de archivos. HbbTV usa el carrusel del objetos.

Cada carrusel de objetos está formado por un directorio raíz que se divide en distintos módulos, cada uno de los módulos transportará uno o más archivos/directorios. Los archivos contenidos en un módulo pueden pertenecer a cualquier parte de la estructura de directorios, no tienen por que pertenecer al mismo directorio. Cuando ya están todos los datos dentro de los módulos se empiezan a transmitir una tras otro hasta que todos han sido transmitidos, al llegar al último se comienza a transmitir de nuevo el primero (de ahí que se le denomine carrusel). Cuando el volumen de datos es relativamente grande este sistema puede ser poco eficiente, por eso los receptores pueden guardar datos en caché. Los receptores pueden hacer caché a nivel de módulo o a nivel de archivo.

HbbTV impone algunas condiciones de uso a la hora de montar y actualizar un carrusel.

- El terminal montará como máximo un solo carrusel y pondrá a disposición de la aplicación la última versión de los archivos del carrusel. Normalmente para montar un carrusel habrá que cachear los datos y luego monitorizar los cambios.
- Para una aplicación bc-related cuya página inicial provenga del broadcast se deberá montar el carrusel, a no ser que para poder montar el carrusel haya que sintonizar otro servicio, en cuyo caso no se cargará la página. Si la página de inicio de la aplicación bc-related no se obtiene del carrusel pero algún elemento de la página si debe ser descargado del carrusel (con un XMLHttpRequest p.ej) se montará el carrusel para que esto pueda llevarse a cabo. Una aplicación bc-independent no podrá montar un carrusel sin convertirse antes en bc-related.
- Durante el ciclo de vida de la aplicación, para una aplicación bc-related, cuando un carrusel haya sido montado cualquier petición que implique montar otro carrusel distinto se realizará desmontando previamente el primer carrusel y cancelar las operaciones pendientes.



4.5.3.2.1 Stream events

Para el radiodifusor el carrusel de objetos DSM-CC es un componente que le permite el envío de aplicaciones y datos independientemente de si el terminal está conectado a la red de banda ancha. Este método de transporte de datos va a ser generalmente la segunda opción de transporte para las aplicaciones interactivas, aunque no debería ser dado de lado pues ofrece todo un canal de comunicaciones que, aunque limitado, es inmediato y puede servir para el transporte de aplicaciones más básicas. Pero es un componente que no ofrece ninguna característica especial en cuanto a la generación de contenido o experiencia especialmente interactiva.

Para conseguir eso DSM-CC pone a disposición del radiodifusor los denominados stream events. Los stream events son unos descriptores embebidos en un flujo elemental DSM-CC de un stream MPEG-2. Con ellos es posible que el receptor se sincronice con el contenido en distintos puntos decididos por el radiodifusor. Por ejemplo el radiodifusor podría usar un stream event para avisar de que está empezando un programa en otro canal de su múltiplex.

Los stream events se clasifican en dos grupos, los objetos y descriptores. Veamos las definiciones para cada uno de ellos:

- Stream event: Es el evento que se dispara en el receptor (el acto) cuando se recibe un descriptor de stream event.
- Objeto Stream event: Es el objeto de Stream event que se transporta en el carrusel DSM-CC. Es como si fuese un descriptor de alto nivel.
- Descriptor de Stream event: Son los descriptores transportados por el flujo DSM-CC que disparan los stream event en la práctica.

Los stream event se almacenan en un carrusel de objetos al igual que cualquier otro objeto DSM-CC. El objeto stream event da información general acerca del stream event: un ID de evento que debe ser único y un nombre para el stream event.

Por su parte, el descriptor del stream event le dice al receptor que se ha generado un evento. El descriptor contiene información más detallada acerca del stream event. La información que contiene un descriptor es: el ID del evento, información temporal acerca de cuando debe dispararse el evento y, la más importante para HbbTV, la capacidad de decirle al receptor que ejecute el evento inmediatamente (“do it now”).

Esta última propiedad de los descriptores de stream event es la que permitirá señalar eventos inmediatos en programas en directo, o señalar los goles de un partido de fútbol, o dar la respuesta a una pregunta de un concurso interactivo a la vez que en el programa, etc...

Los descriptores de stream event que usan el campo “hazlo ahora” solo se transmiten una vez por evento así que el receptor tiene que encargarse de recibirlos correctamente.

Cada vez que el terminal recibe un descriptor de stream event lleva a cabo los siguientes pasos:

- 1 – Comprueba que haya un objeto de stream event con el mismo ID presente en el carrusel. Si no lo hay descarta el descriptor de stream event.
- 2 – Si se detecta que el descriptor es del tipo “hazlo ahora” se dispara inmediatamente.
- 3 – Si el evento no es del tipo “hazlo ahora” se comprueba la referencia temporal para ver cuando debe ser disparado. Si ya existe un evento con el mismo ID listo para ser disparado o si la referencia temporal se ha quedado desfasada, se descarta el descriptor.
- 4 – Cuando la referencia temporal del descriptor alcanza el valor programado se dispara el stream event.

La separación de Stream Events entre objetos y descriptores permite, como hemos dicho antes, la reutilización de los eventos. Es decir, distintos descriptores de eventos con un mismo ID podrían hacer referencia al mismo tipo de evento. Así, siguiendo con el ejemplo del fútbol, un radiodifusor podría decidir usar un mismo ID para lanzar stream events de goles, otro ID para penaltis y otro para finales de partido. Utilizado de esta forma el ID de evento sería el equivalente a una clase de evento.

Al igual que con DSM-CC HbbTV adopta solo ciertas cosas de los Stream Events. Concretamente HbbTV incorpora la funcionalidad “hazlo ahora” de los Stream Event y la que no incorpora es la de sincronizar el Stream Event con un punto temporal predefinido. El ETSI TS 102 809 se refiere a estos últimos como DVB Timeline Stream events. Los radiodifusores deberán ubicar todos los descriptores de stream event de tal forma que puedan ser monitorizados simultáneamente por la aplicación en el mismo PID. Este PID será el mismo que se use para otras secciones DSM-CC.

Para finalizar un gráfico donde se representa conceptualmente lo que significa un carrusel de objetos y los stream events:

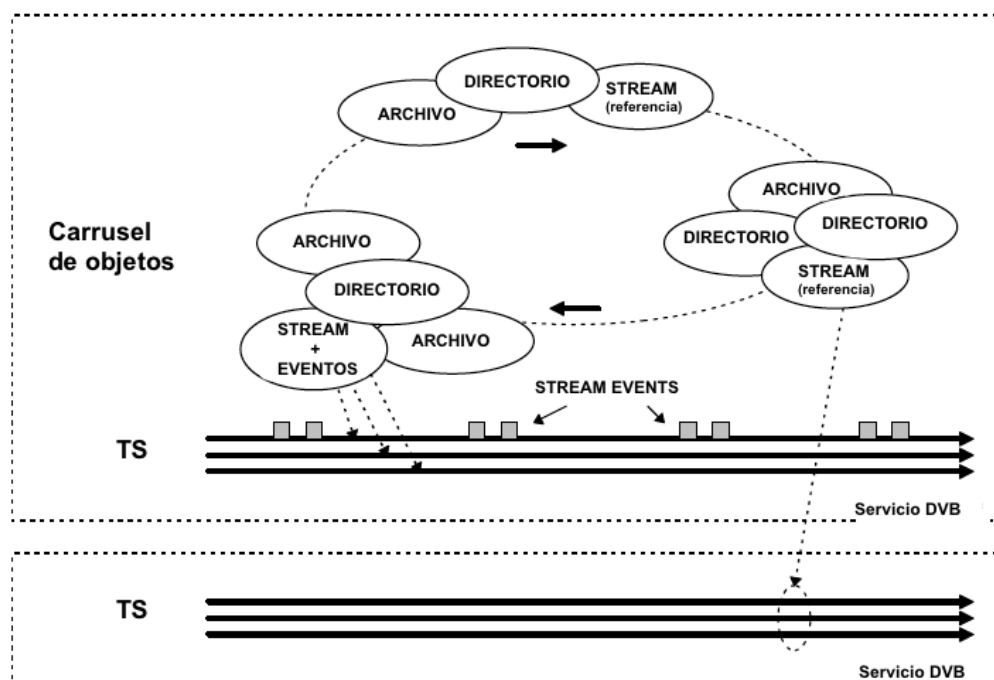


Figura 4.13 – Ejemplo de como se incluyen carruseles de objetos en la especificación. DVB. ETSI TR 101 202 v1.2.1



4.6 Protocolos

HbbTV además del DSM-CC para el canal broadcast, utiliza varios protocolos para el canal broadband.

4.6.1 Streaming

Usa dos protocolos HTTP y RTSP. HTTP es de obligada implementación mientras que RTSP es opcional.

4.6.1.2 HTTP

Para el streaming unicast se usará HTTP 1.1 según su estándar RFC 2616. Y para la entrega de contenidos al terminal se utilizarán dos encabezados HTTP:

- *Range*: Es un encabezado de tipo petición que se usa en peticiones *GET*. Se utiliza para pedir un rango de bytes de una entidad. Se puede pedir un rango directamente, por ejemplo *bytes=500-999* para pedir del 500 al 999 (incluidos) o indirectamente *bytes=-500* pediría los últimos 500 bytes de la entidad. La entidad en este caso sería el vídeo on demand. Con esto se evita tener que descargar por completo el vídeo y permite hacer *trickplay* (pausa, avance, etc...).
- *Content-range*: Este encabezado se envía junto con el “trozo” de entidad que se ha pedido con el método *GET* y el encabezado *Range*. El *content-range* indica la ubicación de ese trozo respecto al total de la entidad. El encabezado debería indicar también el tamaño total de la entidad, a no ser que por algún motivo no sea posible saberla. La función de este encabezado es la de poder hacer operaciones de búsqueda que permitan a la aplicación posicionarse en cualquier punto del vídeo sin la necesidad de tener que descargarlo previamente. Esto también se conoce como “*pseudostreaming*”.

El terminal puede hacer buffer del contenido (si así lo considera) para conseguir que la reproducción del contenido sea fluida y sin cortes. En caso de hacerlo será de unos 10 segundos por delante de la posición de reproducción. Si la velocidad de descarga es mucho menor que la velocidad de consumo la cantidad de segundos del buffer podrá ser mayor.

También se tiene que soportar el “chunked transfer coding” (codificación de transferencia a trozos). Los “transfer coding” (codificaciones de transferencia) se utilizan para indicar que se ha hecho una transformación en la codificación de una entidad para asegurar que su transporte por la red sea seguro.

Se diferencia de un “content coding” en que el transfer coding es una propiedad del mensaje, no de la entidad original. El chunked transfer coding modifica el cuerpo del mensaje para transmitirlo como un flujo de “trozos”. Cada trozo tiene su propio indicador de tamaño seguido de un trailer opcional con los campos de cabecera de la entidad.

Este tipo de codificación permite que se pueda transferir contenido generado dinámicamente junto con la información suficiente para que el que recibe el contenido sepa cuando ha recibido el mensaje completo.



Esto sería un pseudo-código para decodificar una transferencia de este tipo (ejemplo 19.4.6 del RFC 2616):

```

length := 0
read chunk-size, chunk-extension (if any) and CRLF (CRLF = final del
mensaje)
while (chunk-size > 0) {
    read chunk-data and CRLF
    append chunk-data to entity-body
    length := length + chunk-size
    read chunk-size and CRLF
}
read entity-header
while (entity-header not empty) {
    append entity-header to existing header fields
    read entity-header
}
Content-Length := length
Remove "chunked" from Transfer-Encoding
    
```

Para manejar el streaming HTTP se utilizarán los siguientes métodos:

Método	Procedimientos
<i>play(velocidad)</i>	Velocidad es un valor en coma flotante que indica la velocidad de reproducción. El valor 1 indica velocidad real. El valor de velocidad se le pasará al reproductor de medios.
<i>stop()</i>	
<i>seek(posicion)</i>	El reproductor intentará reproducir desde la posición <i>posicion</i> para ello puede usar el encabezado RANGE.
<i>play(0)</i>	Equivale a la pausa.
Propiedad	Procedimientos
<i>read/write String data</i>	Contiene la URI HTTP del elemento del contenido.
<i>Readonly Number playPosition</i>	Contiene la posición de reproducción de la media referenciada por la propiedad <i>data</i> en milisegundos
<i>Readonly Number playSpeed</i>	Se usa para que el terminal indique que velocidades de reproducción es capaz de ejecutar.
<i>Readonly Number playTime</i>	Dice la duración total en milisegundos del archivo a reproducir.
<i>Readonly Number playState</i>	
<i>Readonly Number error</i>	
<i>Readonly Number speed</i>	
Evento	Procedimiento
<i>onPlaySpeedChanged</i>	
<i>onPlayPositionChanged</i>	

Tabla 4.3 – Métodos para controlar el streaming HTTP.



4.6.1.3 RTSP/RTP (+RTSP)

Este sistema de streaming comprende los siguientes bloques de protocolos:

RTP

La especificación para el RTP (RFC 1889 actualizado por el RFC 3550) se compone de los siguientes 2 protocolos:

RTP (Realtime Transport Protocol): es un protocolo diseñado para el transporte de audio y vídeo en redes IP. Está diseñado para transportar flujos de datos en tiempo real de extremo a extremo. Para hacerlo eficiente existen unos perfiles y formatos de payload (carga). Por cada tipo de aplicación (audio o video) se define un perfil y unos tipos de payload. Es decir para poder definir como se usa RTP para una aplicación completa es necesario especificar un perfil y unos formatos de payload. En el perfil se dice los codecs que pueden ser usados para codificar los datos que viajarán en el payload, así como su traducción en códigos de payload que habrá que indicar en el campo correspondiente de la cabecera RTP para ese flujo.

RTCP (RTP Control Protocol): Es un protocolo que funciona en conjunto con RTP. Básicamente se encarga del QoS de RTP y de la sincronización entre el audio y video que transporta ya que RTP transporta el video en un flujo y el audio en otro flujo distinto y al llegar al receptor hay que sincronizarlo para que no se vez un desfase entre la imagen y su sonido.

Los flujos de datos A/V van en puertos pares y sus correspondientes datos de control RTCP en el puerto impar siguiente a cada uno de esos puertos pares.

Por tanto un emisor de un flujo A/V usado RTP capturará los datos y los codificará, después los convertirá en tramas y paquetes con sus correspondientes códigos de tiempo y números de secuencia incrementales . El tipo de payload que se le asignará irá en función del perfil en uso. Se transmiten los paquetes a la red y el receptor los captura , detecta paquetes perdidos, y reordena los que hayan llegado fuera de secuencia. Después de decodifican las tramas en función del tipo de payload usado.

Esto puede hacerse usando como transporte TCP o *UDP*.

RTSP (RFC 2326)

Todo lo anterior nos asegura que el flujo de información llegue de un extremo a otro pero no dice como comenzar la sesión. Además estando la sesión en marcha no se permite control sobre el flujo. Para remediar esto se crea el RTSP (Realtime Streaming Protocol), que es un nombre que lleva a la confusión. Más bien debería llamarse Realtime Streaming Control Protocol RSCP o similar. Este protocolo permite controlar el flujo del streaming (pausa, reproducción, velocidad más rápida, más lenta...). Vale para emisiones en directos y para on-demand aunque HbbTV en la versión 1.0 no soporta emisiones en directo. Es posible usarlo junto con un archivo *SDP* (RFC 2327) que se puede descargar por HTTP y que en pocas líneas define cuando flujos de video y audio hay, de que tipo son, donde se encuentran, en que puertos y que tipo de payload usan (entre otros...).

Este sistema ha sido ampliamente usado durante años pero por su configuración de puertos y protocolos era filtrado en muchas ocasiones por routers y cortafuegos. Por ello, se crearon una par de variantes para sortear este problema.



1 – Interleave en TCP: El streaming RTP/RTSP solía usarse transportado en paquetes UDP. En un streaming de audio e incluso de vídeo perder unos pocos paquetes no era significativo comparado con el ahorro en procesado y tamaño de paquetes que se consigue no transportando en TCP. Pero tenía el problema del filtrado en cortafuegos/routers.

Por ello, el protocolo RTSP incluye una función llamada “Interleaving Binary Data” que permite “intercalar” datos binarios junto con los mensajes de control del flujo del RTSP. El flujo intercalado es el propio de A/V RTP y también el de control RTCP. Un ejemplo de comunicación entre cliente y servidor para establecer una sesión Interleaved (Ejemplo del del RFC 2326 punto 10.12 , RTSP):

```
C->S: SETUP rtsp://foo.com/bar.file RTSP/1.0
      CSeq: 2
      Transport: RTP/AVP/TCP;interleaved=0-1

S->C: RTSP/1.0 200 OK
      CSeq: 2
      Date: 05 Jun 1997 18:57:18 GMT
      Transport: RTP/AVP/TCP;interleaved=0-1
      Session: 12345678

C->S: PLAY rtsp://foo.com/bar.file RTSP/1.0
      CSeq: 3
      Session: 12345678

S->C: RTSP/1.0 200 OK
      CSeq: 3
      Session: 12345678
      Date: 05 Jun 1997 18:59:15 GMT
      RTP-Info: url=rtsp://foo.com/bar.file;
               seq=232433;rtptime=972948234

S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}
S->C: $\000{2 byte length}{"length" bytes data, w/RTP header}
S->C: $\001{2 byte length}{"length" bytes RTCP packet}
```

En el texto subrayado y negrita se ve como el servidor indica al cliente que la sesión RTSP va a ser usando TCP y además intercalado. Además, declara el uso de dos canales 0-1. Por el canal 0 viajará intercalado el RTP y por el 1 el RTCP. Pero esto sigue sin ser la solución definitiva pues se sigue usando el puerto estándar del RTSP (554) y eso tampoco pasaba todos los routers (principalmente los corporativos). Así que hubo que dar otra vuelta de tuerca.

2 – Túnel HTTP: Ya tenemos el flujo RTSP/RTP con sus perfiles y codecs transportado en TCP pero con un puerto y protocolos que no está permitido en el 100 de los cortafuegos y routers. Para conseguir máximo despliegue se pensó en encapsularlo todo en un túnel HTTP, que por puerto y protocolo sí que era accesible desde casi el 100% de routers/proxies/cortafuegos.

Sin entrar en mucho detalle lo que se tuvo que crear un método con sus metadatos y cabeceras específicos que en esencia aprovecha que los métodos *POST* y *GET* de HTTP permiten transportar grandes cantidades de datos. Esto se traduce en que se acaban haciendo las peticiones RTSP con el método *POST* y recibiendo los datos A/V con el método *GET* pero todo ello en el puerto 80 y usando el protocolo HTTP.



Información más detallada en:

<https://developer.apple.com/quicktime/icefloe/dispatch028.html>

El estándar HbbTV implementa lo anterior permitiendo hacer streaming de contenidos basados en MPEG4/AVC para el vídeo y MPEG/AAC para el audio utilizando RTSP y RTP en un contenedor mp4.

Son requisito los protocolos UPD, RTP y un perfil de RTP llamado “RTP Profile for Audio and Video Conferences with Minimal Control” (RFC 3551) y este perfil se extiende con soporte para los *payloads* de H.264 (RFC 3984) y AAC (RFC 3640). También es requisito que se incluya soporta para el método de intercalado de datos RTSP y para el archivo de sesión SDP.

En la siguiente tabla mostramos los métodos y propiedades disponibles para esta función de streaming:

Método	Procedimientos
<i>play(velocidad)</i>	Velocidad es un valor en coma flotante que indica la velocidad de reproducción. El valor 1 indica velocidad real. El valor de velocidad se le pasará al reproductor de medios. Con el valor velocidad=0 el reproductor pausará la reproducción.
<i>stop()</i>	El terminal inicia el método TEARDOWN del RTSP.
<i>seek(posicion)</i>	El reproductor intentará reproducir desde la posición <i>posicion</i> usando el método PLAY de RTSP.
Propiedad	Procedimientos
<i>read/write String data</i>	Contiene la URI RTSP del elemento del contenido.
<i>Readonly Number playPosition</i>	Contiene la posición de reproducción de la media referenciada por la propiedad <i>data</i> en milisegundos
<i>Readonly Number playTime</i>	Dice la duración total en milisegundos del archivo a reproducir.
<i>Readonly Number speed</i>	Valor en coma flotante que indica la velocidad actual de reproducción de la media referenciada por la propiedad <i>data</i> . EL valor de reproducción a velocidad normal es 1.

Tabla 4.4 – Métodos para controlar el streaming RTSP.

4.6.2 Descarga de contenidos (+DL)

La descarga de contenidos HTTP del canal broadband es opcional. HbbTV no obliga al terminal a incluir esta funcionalidad. Pero de incluirla se hará de tal forma que la descarga de contenidos se haga en un sistema de almacenamiento interno opcional. El terminal opcionalmente puede reproducir el contenido mientras es descargado y si el DRM lo permite, puede hacer también trickplay.



La funcionalidad de descarga de contenidos no va a ser ampliamente usada en las primeras fases de despliegue de HbbTV, entre otros motivos porque el contenido interesante para ser descargado suele estar sometido a fuertes derechos de propiedad intelectual y la inversión no suele compensar a los radiodifusores. Por el contrario, para el contenido de producción propia si puede ser interesante ya que estos contenidos no requieren de licencias costosas pero el problema está en que los terminales deberán tener un dispositivo de almacenamiento interno o posibilidad de incorporarlo y no parece que vayan por ahí las intenciones de los fabricantes en estas primeras fases.

4.6.3 Transporte de aplicaciones

El transporte de aplicaciones por el canal broadband se puede hacer usando los protocolos HTTP (RFC 2616) y HTTP sobre TLS (RFC 2818).

En cuanto a la versión de TLS, debe poder utilizarse por este orden:

1º - TLS 1.2 (RFC 5246) → 2º - TLS 1.1 (RFC 4346) → 3º - TLS 1.0 (RFC 2246)

Si no se implementa la 1.2 deberá estar implementada la 1.1 y sino está implementada ninguna de las anteriores deberá implementarse la 1.0. Pero el estándar recomienda que el fabricante incluya la versión 1.2 puesto que es mucho más segura que las otras dos. Versiones futuras del estándar requerirán que solo se soporte la versión 1.2 y no dará opción a ninguna de las otras dos versiones.

Cuando el transporte de la aplicación se haga utilizando HTTP sobre TLS el servidor de la aplicación puede enviar al terminal una petición de certificado. La implementación que se haga de la pila TLS debe permitir la negociación y envío de certificados de cliente al televisor tal cual se requiere en las especificaciones de TLS 1.2, 1.1 o 1.0. El terminal también debe incorporar la extensión de indicación de la renegociación (Renegotiation indication Extension, RFC 5746) para evitar que sea aprovechada una vulnerabilidad en la renegociación SSL y TLS. El certificado instalado en el cliente (el TV) debe cumplir con lo especificado en el RFC 5280. El estándar HbbTV no dice nada acerca de como deben proporcionarse los certificados necesarios para que funcione el sistema.

El terminal deberá soportar los cifrados de cada una de las versiones de TLS que tenga implementadas, según sus especificaciones.

El uso de HTTP sobre TLS en HbbTV tiene varias ventajas:

- Permite desarrollar plataformas de comercio electrónico seguras.
- Permite acceso a servicios que requieran de un certificado digital.
- Permite al radiodifusor filtrar que terminales acceden a sus contenidos.

Para cualquiera de las funciones es necesario que el terminal tenga instalado un certificado digital. Si además permitiese instalar certificados digitales de usuario el abanico de aplicaciones se ampliaría.



4.6.4 Cabeceras User-Agent para el navegador HbbTV

Cualquier petición saliente que haga una aplicación HbbTV deberá incluir un encabezado de agente de usuario (User-Agent) que incluya:

```
HbbTV/1.1.1 (<capabilities>; [<vendorName>]; [<modelName>];  
[<softwareVersion>]; [<hardwareVersion>]; <reserved>)
```

Donde:

- *<capabilities>*: indica las capacidades adicionales que tenga implementadas el estándar. Por ejemplo:
 - “+DL”: Para indicar que permite descargas.
 - “+PVR”: Para indicar que permite grabación y Timeshift.
 - “+RTSP”: Para indicara que permite el streaming RTSP.
- *<vendorName>*, *<modelName>*, *<softwareVersion>* y *<hardwareVersion>* son opcionales y sirven para:
 - <vendorName>*: Nombre del fabricante del terminal.
 - <modelName>*: Modelo del terminal.
 - <softwareVersion>*: Identifica la versión del firmware de la plataforma.
 - <hardwareVersion>*: Identifica la versión del hardware de la plataforma.
- *<reserved>*: está reservado para extensiones futuras.

El encabezado del agente de usuario puede, opcionalmente, contener más información acerca de la implementación. Se recomienda que, al menos, contenga información del navegador HbbTV.

Un ejemplo de agente de usuario:

```
User-Agent: HbbTV/1.1.1 (+PVR+DL; Engel; EN2000; 9.wk.82; 4.2.3)
```

El dispositivo es un Engel EN2000, con versión de Software 9.wk.82 y versión de hardware 4.2.3. Además permite la descarga y la grabación de contenidos.

este también es válido:

```
User-Agent: HbbTV/1.1.1 (;;;;)
```

No tiene características opcionales y no especifica ninguno de los campos que son opcionales



4.7 Aplicaciones y servicios

4.7.1 Introducción

El estándar HbbTV ETSI TS 102 796 v1.1.1 impone que solo haya una aplicación visible a la vez. Esto quiere decir que al pasar de una aplicación a otra la aplicación que deja de estar en primer plano cesa en su ejecución.

Esta es una de las grandes diferencias con la especificación del OIPF DAE. En el DAE si se permite más de una aplicación funcionando al mismo tiempo, de forma que podríamos tener en una parte de la pantalla una aplicación de redes sociales y en la parte de abajo una aplicación directamente relacionada con el programa que se está emitiendo.

Esto no quiere decir que con HbbTV no pueda hacerse eso, si que es posible, pero tendrá que hacerse en el contexto de una sola aplicación.

Como ya hemos mencionado en partes anteriores del documento en HbbTV hay dos tipos de aplicación:

- Bc-related: Van señalizadas en la AIT del canal broadcast y siguen los ciclos de vida que vamos a definir en este punto.
- Bc-independent: O bien no están señalizadas en absoluto o de estarlo será con un archivo XML AIT que será descargado por HTTP como explicamos al final del punto 4.5.3.1 de este documento.

Las aplicaciones pueden hacer una transición de uno a otro tipo como veremos más adelante. Opcionalmente un terminal puede precargar o renderizar aplicaciones en segundo plano, pero no es un requisito de la especificación. En cualquier caso, como hemos comentado al principio, esa aplicación precargada en segundo plano pasará a primer plano tras la interrupción de la aplicación en ejecución.

Tampoco hay ningún mecanismo definido para que las aplicaciones visibles interactúen con otras aplicaciones que estén en ejecución.

Las aplicaciones específicas (locales) del terminal (las que se suelen ver al cambiar de canal, al consultar la EPG, al sintonizar canales, etc...) pueden sobrepresionarse sobre las aplicaciones HbbTV pero esto no les afectará. Si la aplicación local del terminal toma el foco, la aplicación HbbTV que se estaba ejecutando dejará de recibir eventos de teclas del mando a distancia.

Las llamadas a `application.show()` mientras esté visible una aplicación local del terminal:

- harán que la aplicación HbbTV quede detrás de la aplicación local del terminal o
- harán que la aplicación HbbTV vuelva a ser visible si se llamó a `application.hide()` cuando la aplicación específica del terminal se pare.



4.7.2 Ciclo de vida de la aplicación

Vamos a ver como se inician y como se paran las aplicaciones. Que ocurre con las aplicaciones al sintonizar un canal nuevo y que puede ocurrir mientras permanecemos en el canal. Veremos como se gestiona cuando una aplicación está señalizada como broadcast y broadband a la vez y un ejemplo del ciclo de vida de la aplicación.

4.7.2.1 Inicio y fin de las aplicaciones

Una aplicación puede iniciarse de las siguientes formas:

- El usuario la inicia directamente (con un botón del mando a distancia).
- Automáticamente (la aplicación está señalizada como “AUTOSTART” y se inicia al sintonizar el canal en respuesta a la señalización).
- Una aplicación en ejecución utiliza el método JavaScript `createApplication()`

Por defecto, las aplicaciones bc-related no serán visibles para el usuario en el arranque. Deberán llamar a `application.show()` para que el interfaz sea visible y entonces pueda comenzar a recibir ordenes por parte del usuario. Las aplicaciones bc-independent serán visibles y estarán activas sin necesidad de llamar a `application.show()`.

No se permite iniciar una aplicación desde la reproducción de contenido grabado o descargado.

Una aplicación se finaliza de las siguientes formas:

- El usuario la finaliza manualmente.
- Siguiendo lo definido en los diagramas de flujo de la siguiente sección.
- Llamando a la función `application.destroyApplication()`.
- Por errores del terminal.
- Por que el terminal se queda sin recursos y la cierra para seguir funcionando.

Cuando una aplicación quiera iniciar otra, la primera debe continuar ejecutándose hasta que el documento HTML inicial de la segunda se haya cargado por completo. Esto se puede controlar generando un manejador de evento para `onApplicationLoadError` que en caso de ser negativo indicaría que la nueva aplicación ha cargado al menos su página principal, entonces puede cerrarse la primera aplicación.

4.7.2.2 Comportamiento de la aplicación al sintonizar un canal

El siguiente diagrama de flujo indica que ocurre al sintonizar un canal. Al sintonizar un canal se entiende que se contempla también que se puede estar abandonando otro.

El diagrama de flujo no será de aplicación cuando se seleccione un programa MPEG que no sea servicio de broadcast y que no incluya una tabla AIT. Por ejemplo, un servicio MPEG no broadcast es uno que no esté listado en la tabla SDT del TS que las transporta o en un TS que no incluya tabla SDT. La tabla SDT (Service Description Table) listan los proveedores de cada servicio dentro de un Transport Stream. Los datos de la SDT describen los servicios del sistema como los nombres de los servicios, quien proporciona el servicio, etc...



Al seleccionar dicho programa, no broadcast:

- No se iniciará ninguna aplicación.
- No se parará ninguna aplicación a no ser que la aplicación sea bc-related y tenga el descriptor *service_bound_flag=1*. En cuyo caso la aplicación terminaría al cambiar de canal.
- Las propiedades *currentChannel* y *ApplicationPrivateData.currentChannel* indicarán cual es el programa seleccionado.

El descriptor *service_bound_flag* sirve para indicar si una aplicación tiene “continuidad” entre canales o no. Dicho de otra forma, indica si la aplicación está restringida a funcionar únicamente dentro del ámbito de un canal en concreto y debe finalizar al cambiar a otro canal distinto. Con el valor 1 se dice que la aplicación está limitada al servicio.

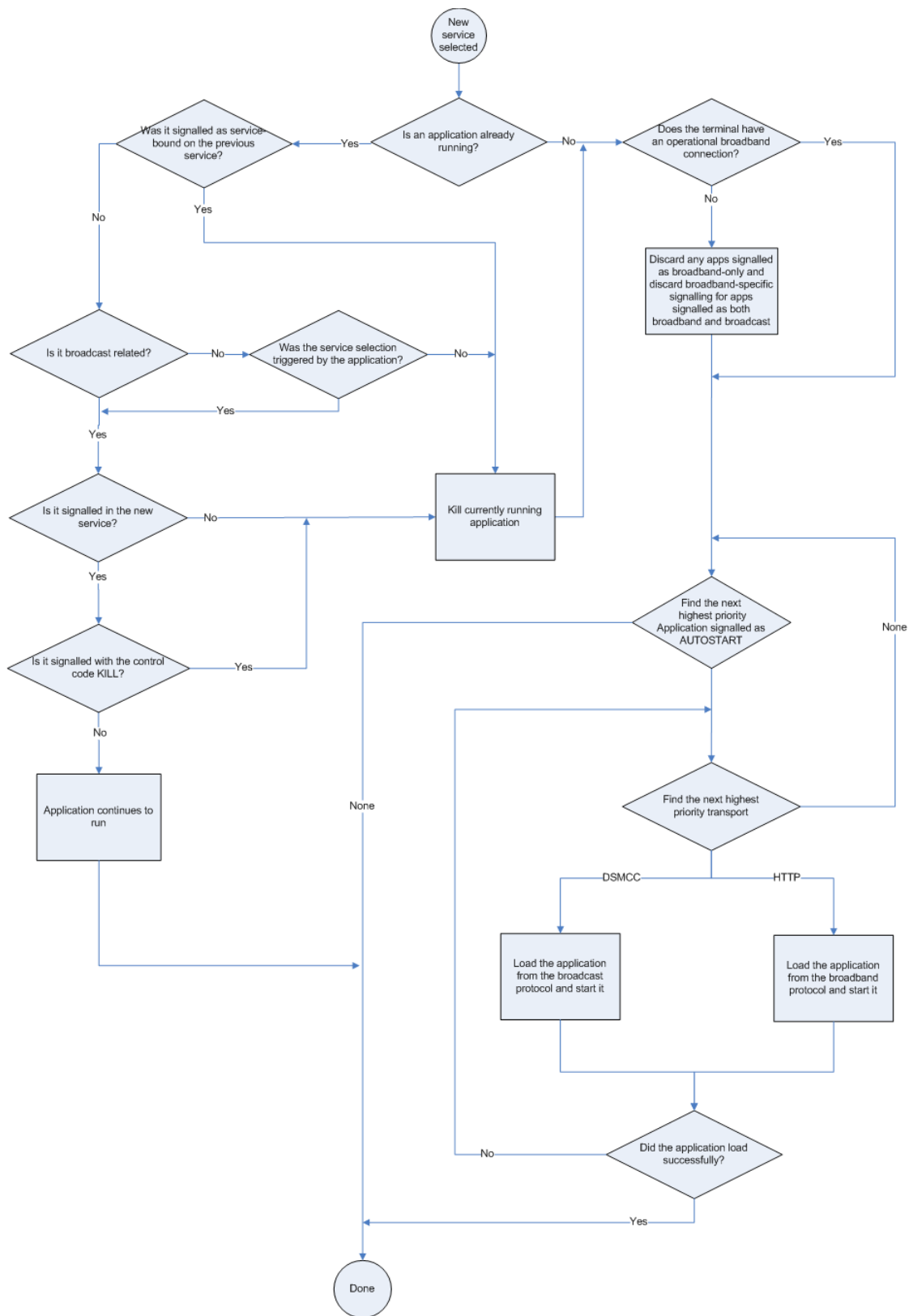


Figura 4.14 – Comportamiento que sigue la aplicación al sintonizar un nuevo canal. ERRATA HbbTV 1.0.

4.7.2.3 Comportamiento mientras se permanece en el canal

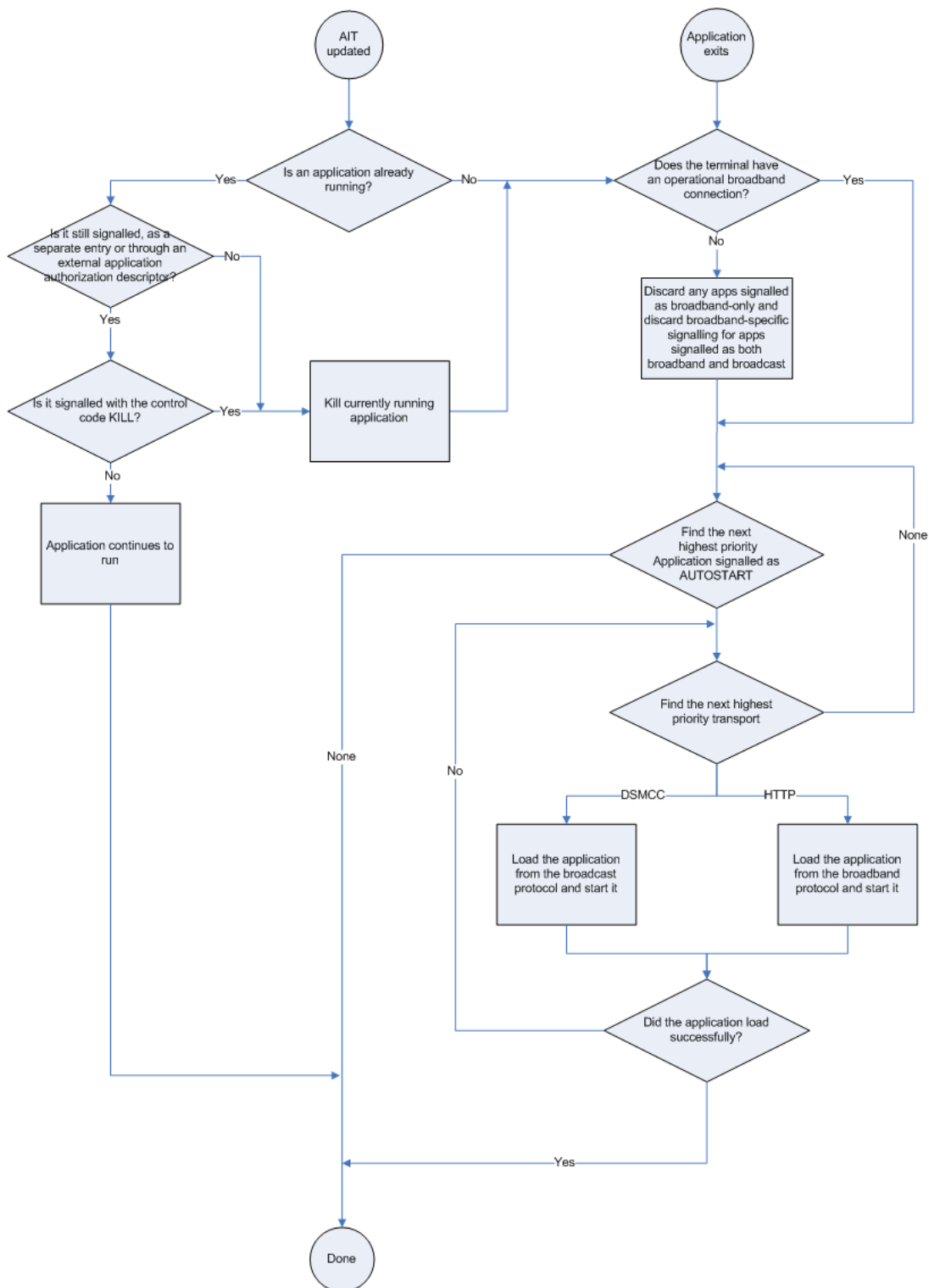


Figura 4.15 – Comportamiento que sigue la aplicación mientras permanece en el canal. ERRATA HbbTV 1.0.



En cuanto a las actualizaciones de la AIT mientras estamos en el mismo canal:

En el bloque de decisión “Is an application already running?” se decidirá si la aplicación está señalizada o si está en ejecución utilizando los campos *organization_id* y *application_id* de la tabla AIT.

Además del *organization_id* y el *application_id* el único campo relevante en una actualización de la AIT es el código de control de la aplicación. Los cambios en el resto de campos no se tienen en cuenta.

Como resultado de lo anterior, los cambios en campos que no sean el *organization_id*, *application_id* o el código de control de la aplicación solo se procesarán para aplicaciones de nueva ejecución. Si se desea que esos cambios afecten a la aplicación en ejecución, habrá que finalizarla y volver a arrancarla.

Un cambio en el número de versión de una sub-tabla de la AIT le dice al terminal que busque una nueva versión de la AIT, aunque esto no implica que haya habido un cambio en el contenido de la AIT. Por ejemplo, al añadirle una aplicación a la AIT podría ser una actualización de la AIT sin tener que cambiar las entradas para las aplicaciones existentes.

Si una aplicación bc-related finaliza sin haber arrancado una bc-independent o sin que se haya cambiado de canal, se volverá a leer la AIT y si hay alguna aplicación marcada como AUTOSTART se arrancará, aunque sea la misma.

Al igual que en el caso anterior, este diagrama de flujo no aplica para servicios MPEG que no sean broadcast.

Los terminales contemplan la opción de arrancar una aplicación de teletexto HbbTV pulsando una tecla TEXT o similar en el mando a distancia. En caso de pulsar el botón TEXT se finalizará la aplicación que está actualmente en ejecución siempre y cuando la aplicación de teletexto esté así identificada en la AIT. Las aplicaciones de teletexto se señalizan con el descriptor *application_usage_descriptor* con un *usage_type=1*.

El estándar HbbTV desde su concepción ha buscado que el teletexto sea una aplicación diferenciada de las aplicaciones AUTOSTART de la AIT. Aún así es posible iniciar la aplicación del teletexto desde una aplicación AUTOSTART sin usar el botón del mando a distancia.



4.7.2.4 Señalización simultánea de aplicaciones en bcast y bband

Prioridades

Se puede señalar simultáneamente la presencia de una aplicación para ser entregada por broadcast y por broadband.

La que vaya señalizada en primer lugar se considera que es la que el broadcaster prefiere como mejor opción pero solo es una indicación que puede servir como base para elegir el protocolo por el que se va a descargar la aplicación.

La decisión final puede depender de otros factores como configuraciones del terminal o el rendimiento que detecte el terminal en cada una de las conexiones.

Cuando la conexión broadband no está operativa

Si no está operativa la conexión broadband:

- Si la aplicación está señalizada como disponible por broadcast y por broadband el terminal descartará la señalización para el protocolo broadband.
- Si la aplicación solo está señalizada como broadband el terminal deberá ignorar la petición de arranque de la aplicación y devolver un error si se usó `createApplication()` para arrancarla.

Cuando hay conexión broadband pero da un error

En este caso, al acceder a la página principal de la aplicación por broadband da un error, si la aplicación está señalizada como:

- Broadcast y broadband, aunque broadband sea la prioritaria se pasará a descargar por broadcast.
- Solo disponible por broadband, el terminal no debe de mostrar ningún mensaje de error en el caso de que la aplicación fuese del tipo AUTOSTART o si la aplicación que falla ha sido arrancada por otra aplicación.

Si la aplicación acaba sin poder ser cargada por ninguno de los dos canales y la aplicación había sido arrancada por `createApplication()` se deberá generar un `ApplicationLoadError`. El estándar HbbTV no dice nada al respecto de las aplicaciones que no arrancan tras ser iniciadas por otras aplicaciones.



4.7.2.5 Aplicaciones bc-independent

Las aplicaciones bc-independent pueden ser creadas de unas de las siguientes formas:

- Llamando a `Application.createApplication()` con la URL HTTP o HTTPS. La URL puede ser una página HTTP/S o un archivo AIT XML como hemos visto anteriormente.
- Opcionalmente, si lo soporta el terminal, con el botón dedicado del portal InternetTV o similar o introduciendo manualmente la dirección en el navegador.

Si la aplicación se crea en base a una URL no se usarán los descriptores `org_id` ni `app_id` y se considerará que el dominio de la aplicación es el *FQDN* de la página HTML, concretamente de la primera página a cargar. Si la aplicación se crea partiendo del AIT XML, se le asignará el `org_id` y `app_id` que indique el AIT XML. El dominio de la aplicación también se indica en el AIT XML.

Cuando una aplicación bc-related inicie una bc-independent la aplicación bc-independent perderá el acceso al servicio broadcast y a sus recursos.

Una aplicación bc-related se puede transformar en una bc-independent llamando a `setChannel()` con el valor `null` para el argumento `channel` y al igual que antes perderá acceso al servicio broadcast y a sus recursos. Si la aplicación quiere volver a ser bc-related deberá almacenar el canal en el que se encontraba antes de convertirse en bc-independent.

En el caso de que una aplicación bc-independent, que no ha sido creada por una aplicación bc-related ni que viene derivada de la conversión de una bc-related en bc-independent seleccione un servicio de broadcast, la aplicación se finalizará, a no ser que cumpla todas las condiciones siguientes:

- La aplicación tiene `org_id` y `app_id` que ha obtenido bien por un AIT broadcast o XML.
- En el servicio broadcast seleccionado hay señalizada como AUTOSTART o PRESENT una aplicación con el mismo `org_id` y `app_id` que incluye un `transport_protocol_descriptor` con el valor `protocol_id=3`. (Es decir, HTTP por canal broadband como decíamos en el punto 4.5.3.1 de señalización AIT).
- La URL del documento de entrada de la aplicación bc-independent coincide exactamente con la URL para el `org_id` y `app_id` señalizados.
- La página de la aplicación bc-independent cargada en ese momento pertenece al dominio de aplicación de la aplicación bc-related.

Si se cumplen todas las condiciones, la aplicación puede hacer la transición a bc-related y comportarse como una bc-related original.

Como se puede ver las condiciones por las cuales una aplicación bc-independent se pueda convertir en una bc-related son muy restrictivas y no sería posible satisfacerlas todas si no es con la colaboración del radiodifusor, que debe señalizar en el canal broadcast lo necesario para que se puedan cumplir las condiciones.

Posibles usos de esta funcionalidad serían que desde una aplicación de portal del fabricante se pudiese sintonizar el canal del broadcaster y comenzar a usar los servicios definidos para el canal. Un ejemplo similar sería estar navegando por la web del broadcaster en el navegador del terminal y sintonizar el canal de broadcast para el mismo fin.

4.7.3 Ejemplo del ciclo de vida de una aplicación

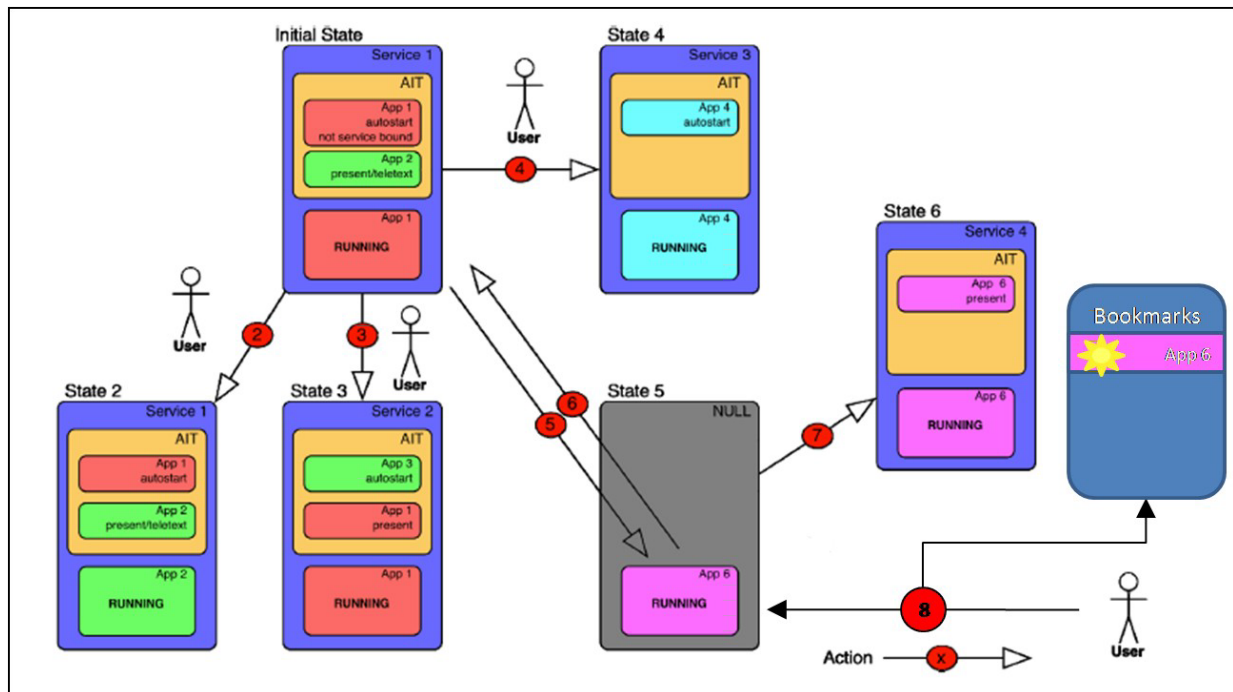


Figura 4.16 – Combinaciones para el ciclo de vida de una aplicación. ERRATA HbbTV 1.0.

Estado inicial	Nº Acción	Estado final
Initial State → La App 1 se está ejecutando	2 → El usuario pulsa "TEXT"	State 2 → Se inicia la App 2
Initial State → La App 1 se está ejecutando	3 → El usuario cambia al canal 2	State 3 → La App 1 sigue ejecutándose dado que no es service-bound (limitada al canal)
Initial State → La App 1 se está ejecutando	4 → El usuario cambia al canal 3	State 4 → Se finaliza la App 1 y se arranca la App 4 que es AUTOSTART en su canal.
Initial State → La App 1 se está ejecutando	5 → La App 1 hace una llamada createApplication() con un XML AIT para iniciar una aplicación bc-independent.	State 5 → Se ejecuta la App 6. Se pierde el acceso al canal broadcast y a sus recursos. La App 6 tiene un <i>app_id</i> , ya que ha sido iniciada por un XML AIT (Punto 4.7.2.5)
State 5 → La App 6 está ejecutándose	6 → El usuario cambia al canal 1	State 1 → Se termina la App 6 y se inicia la App 1 que es AUTOSTART.
State 5 → La App 6 está ejecutándose	7 → La App 6 sintoniza el canal 4	State 6 → Se reproduce el contenido del canal 4 y como en él se está señalizando la App 6, ésta se convierte en bc-related y no se interrumpe su ejecución. Esto es lo que hemos visto en el punto 4.7.2.5.
	8 → El usuario introduce la URL de un AIT XML (o una dirección) para arrancar una aplicación y la guarda en sus favoritos. El terminal graba el nombre y logo de la aplicación para el favorito.	

Tabla 4.5 – Explicación de los distintos estados de la aplicación de la Figura 4.16.



4.8 Seguridad

El estándar HbbTV define dos niveles de confianza para las aplicaciones: de confianza y de no confianza. Las funciones con nivel de no confianza no pueden acceder a funciones del terminal.

Por defecto, las aplicaciones bc-related se considerarán de confianza y la bc-independent de no confianza. Este comportamiento puede verse modificado por lo siguiente:

- El terminal puede permitir que el usuario marque aplicaciones bc-independent como de confianza y que marque aplicaciones bc-related como de no confianza para algún servicio o canal.
- Si el terminal puede recibir canales no regulados, no debería aceptar las aplicaciones bc-related como de confianza. HbbTV no determina que considera como un canal no regulado.
- El estándar HbbTV no obliga a que todas las aplicaciones bc-related sean consideradas de confianza. Puede ser que por regulaciones locales ciertos tipos de aplicaciones bc-related deban considerarse de no confianza.
- El fabricante también puede marcar aplicaciones bc-independent como de confianza y aplicaciones bc-related como de no confianza.

Certificados de usuario

Si se usa HTTP sobre TLS es necesario un certificado digital de cliente para que el proveedor del servicio pueda autenticar al cliente. Algunos modelos de negocio requieren que la aplicación HbbTV se distribuya exclusivamente a terminales HbbTV certificados.

En esta versión del estándar no se incluye ningún mecanismo por el cual expedir o incluir certificados digitales de cliente al terminal, pero se está trabajando en ello para futuras versiones.

El uso del certificado nos parece interesante para poder ofrecer servicios públicos a los ciudadanos en base a su certificado digital, por ejemplo, el de la *FNMT*. Pero para ello, primero debe existir un mecanismo capaz de depositar el certificado en el terminal y el terminal debe ser capaz de autenticarlo para lo que debe tener instalado el certificado de la autoridad certificadora (CA) que lo expide, en este caso la *FNMT*.

HbbTV obliga a incorporar una serie de CAs y ninguna de ellas es el de la *FNMT*.

Por ello, desde RTRM hemos pedido incorporar en la especificación nacional de interoperabilidad que los terminales incluyan a la CA de la *FNMT* entre sus CAs preinstaladas, a falta de que se desarrolle el mecanismo para poder depositar la firma digital personal en el terminal.

A nivel nacional, en lo que se está trabajando es en utilizar el certificado digital del dispositivo para autenticar si el terminal cumple con los requisitos de interoperabilidad del mercado nacional, actualmente en desarrollo. Cada terminal que pida una aplicación HbbTV presentaría su certificado digital y el proveedor de la aplicación o cruzaría con una base de datos con listas blancas o negras de acceso.



4.9 Capacidades y funciones del terminal

Estas son las capacidades mínimas que el terminal debe ofrecer a las aplicaciones HbbTV. Se pueden añadir capacidades adicionales en el perfil de capacidades XML.

4.9.1 Capacidades mínimas

A diferencia de otras secciones del estándar, en este caso no se incorpora ninguna de las del OIPF DAE, son definidas directamente por el estándar HbbTV. Son las siguientes:

- Resolución del plano gráfico HbbTV: 1280x720 píxeles con una relación de aspecto de 16:9. El terminal deberá tener esta resolución como mínimo, si la resolución es mayor a las aplicaciones se les seguirá reportando como si fuese 1280x720.
- Formato de color: RGBA32 para resoluciones menores al menos debe soportar RGBA4444. Se permite la superimpresión de video (video overlay).
- Fuente proporcional, como mínimo la fuente “Tiresias Screefont v8.03” (o equivalente) con la codificación de caracteres Unicode “Basic Euro Latin Character Set” tal cual está definido en el anexo D del ETSI TS 102 809. Esta fuente debe poder ser accesible desde un CSS con la regla: `font-family: Tiresias;`
Como curiosidad, decir que la fuente Tiresias fue desarrollada por la BBC para ser usada con sus aplicaciones interactivas. Es una fuente especialmente diseñada para las pantallas de TV. Es una fuente sans-serif (sin ribetes), escalable y con anti-alias. Dos fuentes son “equivalentes” cuando tienen la misma altura, la misma anchura, tienen los mismos caracteres y son visualmente parecidas.
- Fuente no proporcional, “Letter Gothic 12 pitch” (o equivalente) en las mismas condiciones que la anterior y accesible por CSS con: `font-family: “Letter gothic 12 Pitch”;` Es una fuente mono espacio, escalable y con anti-alias.
- Método de entrada de texto: Debe incorporar el método multi-tap (tipo SMS de cuando los teléfonos solo tenían teclas) o tipo teclado software. En el caso del multi-tap o métodos equivalentes que usen eventos de teclado para introducir texto, los eventos no se pasarán a la aplicación, solo la cadena de texto final es la que se pasará a la aplicación. Desde el CSS se puede usar el método `input-format` para determinar cual de los métodos se va a usar.
- Número mínimo de filtros para DSM-CC: El terminal debe reservar recursos suficientes para registrar secciones DSM-CC para 3 flujos elementales (elementary streams) simultáneos, para un mismo carrusel DSM-CC. Y debe reservar al menos otro filtro para poder monitorizar los Stream Events de ese DSM-CC.
- Cache para el DSM-CC: Define que como mínimo ha de tener 3MB.
- Streaming unicast HTTP y descargas, formatos de archivo: se permitirán los formatos TS y mp4.
- Streaming unicast HTTP y descargas, formatos de video: los perfiles AVC_SD y AVC_HD.



- Streaming unicast HTTP y descargas, formato de audio: HEAAC, AC3 y MP3.
- Gestión PVR: Si está implementada, el atributo `manageRecordings` deberá tener el valor "samedomain".
- Descargas: Si está implementada, el atributo `manageDownload` deberá tener el valor "samedomain".
- No se requiere la presentación simultánea de contenido broadcast y broadband.
- Control Parental: Como mínimo ha de soportar la identificación de edad mínima, codificada según EN 300 468.
- Escalado de video: Los terminales deben poder escalar video en un factor de 0.125x0.125 y 0.25x0.25 . Para tamaños entre esos dos valores los terminales que no puedan escalar vídeo deberán recortarlo. Los terminales también deben poder escalar el video en un factor de 2x2. Además de esto, se podrá escalar video de forma arbitraria siempre y cuando se respete la relación de aspecto del video de tal forma que todo el video quepa en el área del objeto video/broadcast.
- Cookies: Las cookies que tengan fecha de expiración deberán almacenarse en la memoria persistente y los terminales deberán respetarla. Se soportarán al menos 100 cookies con un máximo de 20 por dominio y un tamaño individual hasta 4KB.

4.9.1.1 Acciones de usuario

El usuario debe poder manejar las aplicaciones con acciones de las teclas del mando a distancia. En la siguiente tabla se listan los tipos de evento de teclado mínimo que debe tener implementado cualquier terminal.

Botón	Evento de teclado (key event)
Rojo, azul, verde, amarillo	<code>VK_RED, VK_GREEN, VK_BLUE, VK_YELLOW</code>
Flechas arriba, abajo, dcha, izq.	<code>VK_UP, VK_DOWN, VK_RIGHT, VK_LEFT</code>
ENTER / OK	<code>VK_ENTER</code>
ATRÁS / BACK	<code>VK_BACK</code>
Números (0 – 9)	<code>VK_0 → VK_9</code>
Play, Stop, Pause	<code>VK_PLAY y VK_PAUSE o VK_PLAY_PAUSE, VK_STOP</code>
Avance/Retroceso rápido	<code>VK_FAST_FWD, VK_REWIND</code>
TEXT o TXT	Este no se pasa a las aplicaciones, el terminal lanza la aplicación del teletexto directamente.

Tabla 4.6 – Eventos de teclado obligatorios en HbbTV.

NOTA: Lo de VK viene del CEA2014, y significa Virtual Key.

Las key events anteriores estarán disponibles para las aplicaciones cuando las pidan mediante un objeto `KeySet`. Las que no estén en la lista las tramitará la implementación del terminal y no se pasarán a la aplicación.



Cuando se esté utilizando la navegación de teclas arriba, abajo, izquierda, derecha, el terminal decidirá cual de los siguientes mecanismos utilizar, ordenados por prioridad:

- Permitirá a las aplicaciones capturar los eventos evitando la llamada “navegación JavaScript”.
- Utilizar la navegación direccional del CSS3 *nav-up*, *nav-right*, *nav-down* y *nav-left*.
- Usar la navegación del terminal que permite ir pasando por todos los elementos que puedan tener foco, conocida como “navegación por foco”.

De todas formas las aplicaciones deberán marcar el bit NAVIGATION del objeto KeySet aunque las teclas de navegación se usen solo para hacer navegación por foco y no se usen para navegación javascript ni por CSS con *nav-**.

4.9.2 Funciones del terminal

Favoritos

El terminal debería proporcionar un mecanismo para organizar las aplicaciones bc-independent más usadas. Cuando se dice debería, no implica obligación pero si que debe haber un motivo muy claro por el cual no hacerlo.

Para las aplicaciones los portales del fabricante o de listas de favoritos (*www.hbbig.com*, por ejemplo) el terminal puede usar el título, el icono que se especifica en la sección de encabezado HTML y la URL de la primera página.

Streaming y Descarga

Los terminales no permiten que se guarde el contenido streaming en una memoria persistente cuando ese contenido se ha iniciado usando el API de streaming (CEA 2014 AV Control Object). Si el proveedor de contenidos quiere que el contenido pueda ser almacenado, deberá iniciar la descarga utilizando el API de descargas.

Grabación (PVR)

Es el terminal el que decide si ejecutar las ordenes directamente o si permite que sea la aplicación quien pueda realizar acciones, como por ejemplo abrir un diálogo para pedir permiso al usuario para empezar la grabación.

4.9.3 Consulta de capacidades y cadenas opcionales

Es posible consultar al terminal la lista de capacidades que tiene implementadas. Es una forma de saber que funciones hace el terminal para poder adaptar los contenidos a sus capacidades. El intercambio se hace usando archivos XML codificados de tal forma que en un archivo se pasan los detalles de todo lo que el terminal puede poner a disposición de la aplicación.



Para el nivel de funciones básico se usará el siguiente XML como propiedad *xmlCapabilities* del objeto *application/oipfCapabilities*:

```
<profilelist>
  <ui_profile name="OITF_HD_UIPROF+DVB_T+ " />
  <video_profile name="TS_AVC_SD_25_HEAAC" type="video/mpeg" />
  <video_profile name="TS_AVC_HD_25_HEAAC" type="video/mpeg" />
  <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" />
  <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" />
</profilelist>
```

DVB_T puede sustituirse por el sistema de radiodifusión utilizado, o se puede añadir más sistemas si el terminal así lo lleva incorporado.

Como se puede ver, los perfiles de A/V coinciden con los mencionados en el apartado de formatos para el canal broadband (punto 4.5.2.2). No se especifican los formatos del canal broadcast.

Si además el terminal soporta audio E-AC3 el XML sería ampliado con:

```
<profilelist>
  <ui_profile name="OITF_HD_UIPROF+DVB_T+ " />
  <video_profile name="TS_AVC_SD_25_HEAAC" type="video/mpeg" />
  <video_profile name="TS_AVC_HD_25_HEAAC" type="video/mpeg" />
  <video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4" />
  <video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4" />
  <video_profile name="TS_AVC_SD_25_E-AC3" type="video/mpeg" />
  <video_profile name="TS_AVC_HD_25_E-AC3" type="video/mpeg" />
  <video_profile name="MP4_AVC_SD_25_E-AC3" type="video/mp4" />
  <video_profile name="MP4_AVC_HD_25_E-AC3" type="video/mp4" />
</profilelist>
```

En la siguiente tabla vemos cuales son las cadenas válidas para indicar cuales de las funciones opcionales están implementadas en el terminal:

Cadena opcional	Significado
"+DL"	Se permite la descarga de contenidos
"+PVR"	Se permite la grabación de contenidos
"+RTSP"	Se permite el streaming RTSP

Tabla 4.7 – Cadenas para indicar funcionalidades adicionales en el terminal.



Las cadenas definidas en esta tabla se usarán de la siguiente forma:

- En el encabezado User-Agent HTTP
- Como hemos visto arriba en el XML, como opción dentro del *ui_profile name* del objeto *application/oipfCapabilities*.
- Como parámetro del método *hasCapability()* del objeto *application/oipfCapabilities*. Así es como se pregunta de forma dinámica si un terminal tiene una determinada característica implementada.

El soporte para DRM se indicará usando el elemento `<drm>`. Un ejemplo que indica que se soporta el DRM para el TS del canal broadcast:

```
<drm DRMSystemID="urn:dvb:casystemid:12345">TS</drm>
```

El soporte para CI+ se indicará usando también el elemento `<drm>` pero añadiendo el atributo *protectionGateways* con el valor "ci+". Ejemplo que indica que se puede acceder al contenido protegido por DRM con un sistema Common Interface:

```
<drm DRMSystemID="urn:dvb:casystemid:12345"  
protectionGateways="ci+">TS</drm>
```

En HbbTV el soporte de CI+ se hace pensando en el canal broadcast, no en el broadband. Para el broadband está previsto usar DRMs.

4.10 Evolución del estándar

El estándar, desde su publicación, no ha parado de evolucionar. Ya ha sido publicada una errata a la versión 1.0 y la nueva versión 1.5 del estándar. Y ya se está trabajando en la versión 2.0.

La versión 1.5 se espera que sea publicada como estándar este año 2012 bajo el nombre ETSI TS 102 798 v1.2.1. No se sabe la fecha de publicación exacta de la versión 2.0, pero se estima que puede ser en el primer cuatrimestre de 2013. Esta fecha variará casi con total seguridad en función de la aceptación que tenga el estándar en el mercado de consumo.

Vamos a ver las novedades que aportarán las futuras versiones.

4.10.1 HbbTV 1.5 (futuro ETSI TS 102 798 v1.2.1)

HbbTV 1.5 se publicó el 16 de Marzo de 2012. Es una extensión de HbbTV 1.0 (ETSI TS 102 796 v1.1.1). El documento se ha presentado con un poco de premura para intentar estandarizar a nivel europeo algunos conceptos que comienzan a particularizarse para mercados nacionales.

A nivel técnico las novedades más significativas son las siguientes:

- Añade soporte para streaming MPEG DASH
- Un sistema común de encriptación que permite el uso de múltiples DRMs
- Acceso a la información DVB-SI EIT

Con estas incorporaciones la relación con el estándar HbbTV 1.0 queda así:

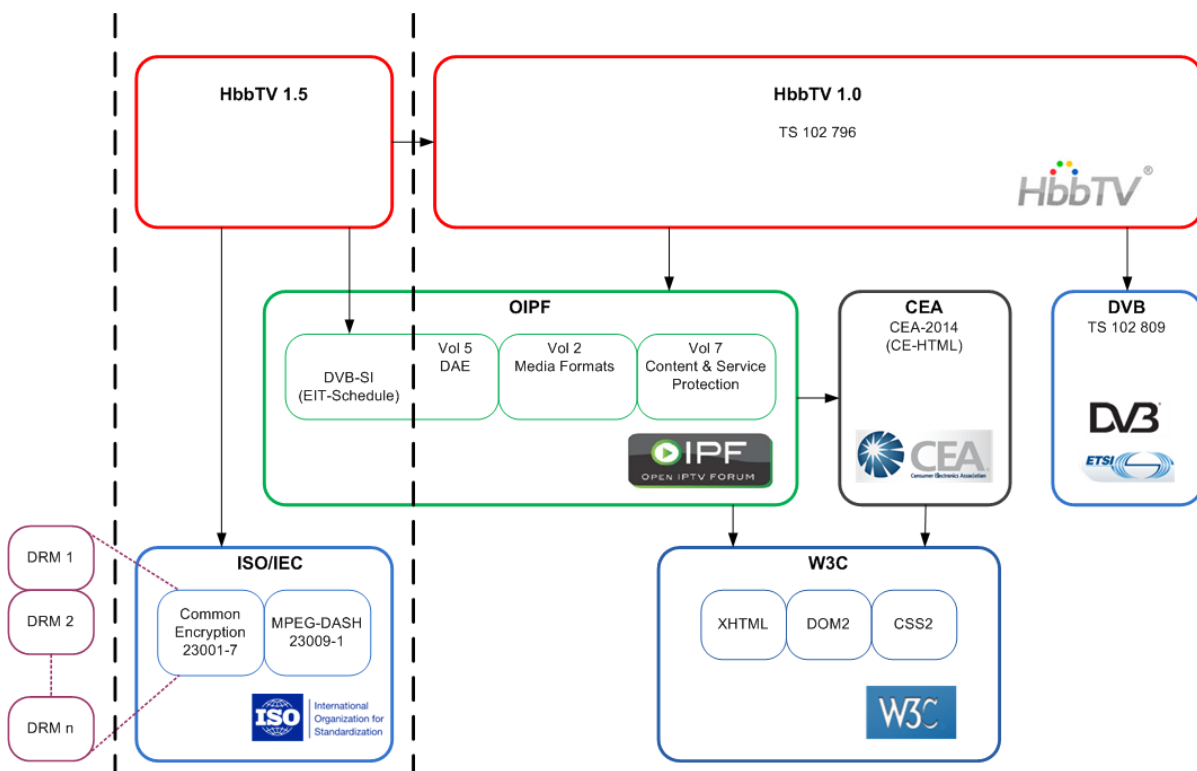


Figura 4.17 – Esquema con las nuevas funcionalidades de HbbTV 1.5.



El documento del HbbTV 1.5 modifica muy pocas cosas del estándar ETSI TS 102 796. Se limita a añadir nuevas características en ciertos puntos del documento, hay otros que quedan intactos. Por ejemplo se añaden nuevas entradas al XML de intercambio de capacidades:

```
<clientMetadata type="dvb-si">true</clientMetadata>
```

para el DVB-SI y

```
<video_profile name="MP4_AVC_SD_25_HEAAC" type="video/mp4"
  transport="dash"/>
<video_profile name="MP4_AVC_HD_25_HEAAC" type="video/mp4"
  transport="dash"/>
```

para el DASH.

Se modifican algunas cosas del Browser Application Environment y se añade la encriptación CENC al apartado de seguridad. También se añaden al final 3 anexos explicando detalles de implementación de las nuevas tecnologías.

De entre todos los elementos que cambian (que son muy pocos) vamos a ver el que consideramos más relevante, que es el MPEG DASH.

4.10.1.1 MPEG DASH (Dynamic Adaptive Streaming over HTTP)

Esta es probablemente la novedad más interesante. Es una estándar muy reciente, su aprobación final como estándar ISO se ha producido en Abril de 2012.

Vamos a introducirlo diciendo que MPEG en los últimos años ha desarrollado varias tecnologías para el transporte multimedia, entre ellas el MPEG-2 Transport Stream (TS) y el ISO Base Media File Format (MP4) de los que ya hemos hablado en este documento. Como ya hemos visto, estas tecnologías son ampliamente utilizadas en la industria del broadcast, el transporte de audio y video por internet, teléfonos móviles, etc... A esto le sumamos que internet hoy día ya es uno de los canales más importantes para la distribución de contenidos y su protocolo estrella es el HTTP, que lleva ya tiempo utilizándose para la entrega de contenidos multimedia.

Lo que no existía es un estándar que uniese estas dos tecnologías, las de transporte (TS, MP4) con las de distribución (HTTP). MPEG lo que ha hecho con DASH es estandarizar el streaming HTTP de contenidos MPEG.

Los objetivos del estándar son:

- La entrega eficiente de contenidos MPEG utilizando HTTP en las variantes: adaptativa (esto es nuevo), progresiva y streaming/descarga.
- La emisión de contenidos en directo.
- Uso de la actual infraestructura de CDNs, Proxys, Cachés, NATs y cortafuegos.
- Señalización, entrega de contenidos, uso de múltiples DRMs

Tradicionalmente el streaming en internet se llevaba a cabo con protocolos como RTSP que son "stateful" (con estado): el cliente se conecta al servidor de streaming y el servidor mantiene la conexión con el cliente comprobando su estado hasta que el cliente se desconecte. Hay mucha

comunicación de control entre el cliente y el servidor.

Más adelante se comenzó a usar HTTP para la entrega de contenidos multimedia. HTTP es un protocolo sin estado: el cliente pide datos al servidor, el servidor los manda y termina la conexión. Si se quiere mantener algún tipo de sesión o control de lo enviado por el servidor al cliente hace falta usar cookies. Como evolución al streaming RTSP, se creó el llamado “pseudostreaming” o “descarga progresiva” que permite usar servidores HTTP 1.1 para la distribución de contenidos. El sistema se basa en ir descargando trozos de contenido para ir reproduciéndolos en orden. Aún así, el cliente puede buscar y reproducir en puntos arbitrarios del contenido, algo que consigue haciendo peticiones HTTP al servidor indicando un *byte-range*. Así lo hacen proveedores de contenidos tan importantes como Youtube.

Sin embargo la descarga progresiva tiene algunas deficiencias, por ejemplo:

- Como se descargan trozos enteros para poder visualizarlo, si el cliente decide dejar de ver el contenido, habrá descargado una cantidad de datos que no va a consumir → ancho de banda desaprovechado.
- Aunque en sitios como Youtube te permiten elegir la calidad de la reproducción, esta forma de hacerlo no es adaptativa, es fija basada en búfer.
- No permite hacer streaming de contenidos en directo.

DASH soluciona todas las debilidades de RTP/RTSP y del pseudostreaming HTTP.

DASH es una solución de streaming cuyo fundamento es que el cliente se descarga pequeños trozos (segmentos en DASH) de video por HTTP que luego junta y reproduce. El cliente controla la entrega del contenido.

Este es un ejemplo de arquitectura de distribución para streaming DASH:

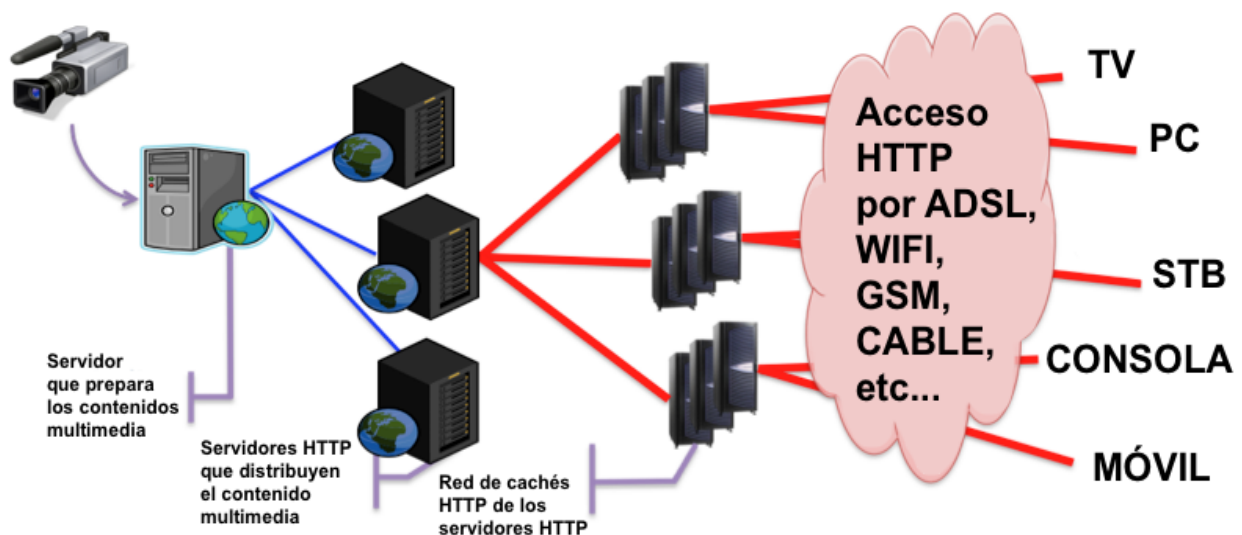


Figura 4.18 – Arquitectura de un sistema DASH. Thomas Stockhammer.



En la Figura 4.18 observamos que el contenido original se deja en un servidor (Media Preparation) que lo divide en segmentos y los codifica en diferentes resoluciones y calidades. Luego, esos segmentos se mueven a distintos servidores (Media HTTP Origin Servers) junto con unos archivos llamados *MPD* que son unos archivos de metadatos con unos “índices” que indican que segmentos componen cada archivo para cada una de las codificaciones que se han hecho. En función de la información contenida en estos MPDs el cliente puede pedir el archivo usando HTTP GET o HTTP GET parciales como hemos visto antes. El cliente controla la sesión de streaming de forma local, pide archivos de más calidad cuando su conexión es más estable y de menos calidad cuando la calidad de la conexión se degrada, pero esto lo hace de forma automática. Los HTTP Cachés es la parte que se reutiliza de las infraestructuras actuales de distribución de contenidos HTTP.

Para distribuir contenidos de forma masiva hacen falta granjas de servidores que puedan responder a los millones de peticiones de clientes que hay, esta infraestructura ya está desarrollada y probada con CDNs, cachés, etc... por eso se ha optado por reutilizarla. Además de que no se elevan los costes de desarrollo para poder poner en marcha DASH ya que todas las inversiones en infraestructuras ya están hechas.

DASH no es un protocolo, codec, middleware o especificación de cliente. DASH se puede entender como un sistema por el cual se proporcionan formatos que habilitan la entrega eficiente y de alta calidad de servicios de streaming. Se considera más un componente dentro de una cadena de distribución punto a punto.

Sus objetivos son:

- Reutilizar la tecnología existente de contenedores, codecs, DRMs, etc...
- Que pueda ser desplegado sobre la infraestructura de CDNs-HTTPS actual.
- Que el usuario perciba un mejor servicio: menor tiempo de arranque, sin bufer...
- Permite seleccionar la calidad en función de la red, capacidad del dispositivo.
- Los cambios entre calidades son automáticos y transparentes al usuario.
- Permite contenidos en directo y con menú y características tipo DVD.
- Puede coexistir con tecnologías propietarias existentes.

HbbTV adopta DASH porque necesita un sistema que sea capaz de entregar contenidos por el canal broadband de tal forma que se reproduzcan con fluidez y sin cortes. Recordemos que los terminales donde se ejecuta HbbTV no suelen tener características técnicas muy potentes y un sistema de este tipo ayuda a que puedan hacer mejor su trabajo. Por ejemplo, en un ordenador si elegimos una calidad superior a la que nuestra conexión nos permite es fácil que cambiemos a una calidad inferior, pero esto en una TV no es tan sencillo, es más, es probable que no se pueda hacer y los contenidos siempre se mostrarán con la misma calidad funcionen o no.

Hay otro punto muy importante por el cual HbbTV quiere incorporar DASH y es que además de la fiabilidad en el entrega del contenido permite la difusión de contenidos en directo por el canal broadband.

DASH también tiene una serie de perfiles que definen distintas calidades y tipos de uso. HbbTV solo usa un perfil de DASH el “ISO Base Media File Format live profile”, este perfil permite streaming en directo y bajo demanda de contenidos ISOBMFF, que en el caso concreto de HbbTV será usando el contenedor MP4. También permite seleccionar distintas pistas dentro del contenido (subtítulos, otros idiomas, etc...).


4.10.2 HbbTV 2.0

Aun no se sabe con certeza lo que va a incluir la versión 2.0 del estándar, pero en distintos foros se ha especulado acerca de lo que puede llevar. HbbTV podría incorporar algunas de las siguientes características:

- Más funciones para el streaming adaptativo (DASH).
- Segunda pantalla (tabletas, móviles).
- Sincronización de flujos broadcast y broadband.
- Gráficos avanzados.
- Incorporación de código HTML5 y nuevas tecnologías Web.
- Integración para contenido 3D.
- Descarga de tipos de fuentes.
- Lanzamiento de VoD desde el canal broadcast.
- Aplicaciones para manejar las grabaciones.
- Widgets.


De todo lo anterior lo que ya es una realidad son las aplicaciones de segunda pantalla. Esto consiste en que el usuario pueda manejar e interactuar con las aplicaciones HbbTV desde otro dispositivo como un teléfono móvil o una tableta. Un ejemplo de uso es, por ejemplo, estar sentado delante del televisor y lanzar la aplicación HbbTV desde la tableta, seleccionar una lista de vídeos para ver, y que cuando termine de ver los vídeos la aplicación recuerde cuales ha visto y la próxima vez que use la aplicación recordarle cuales le faltan por ver. O más aún, acabar de verlos en la tableta en la cocina mientras cena. También puede iniciarse una compra en el televisor y dar los datos de facturación por el móvil. Hay compañías españolas como Abertis que ya lo están desarrollando. En Alemania hay un consorcio denominado HBB-NEXT compuesto por unas 10 empresas (entre ellas el IRT, NEC, TARA Systems,...) que están llevando a cabo un proyecto para el desarrollo de aplicaciones avanzadas para la televisión híbrida. El proyecto incluye servicios adicionales como reconocimiento vocal, gestos para el control de la TV, mecanismos más avanzados de seguridad y control, etc...

News




19-20.06.2012

Course set for Milestone 6
On 19 and 20 June some HBB-NEXT partners met in Munich at TARA systems and IRT to set the course for their work towards the project's Milestone 6. Milestone 6 is to be reached by end of September and envisages a presentation of a first version both



20.06.2012

Broadcast-broadband synchronization client accomplished
HBB-NEXT partner TNO has just completed a first HBB-NEXT synchronization prototype – a real milestone for the projects synchronization stand. ...>>



21.05.2012

HBB-NEXT at IFIP Networking
IFIP (The International Federation for Information Processing) celebrated the 40 year anniversary of its Technical Committee on Communication Systems (TC6) in Prague on 21 May 2012 - HBB-NEXT was present. ...>>

Figura 4.19 – Captura de la página web de <http://www.hbb-next.eu/> donde se puede ver con fecha del 20/6 el desarrollo del cliente de sincronización broadcast y broadband.



BLOQUE 3 - Desarrollo práctico de la especificación

En este apartado vamos a ver como ha sido el desarrollo práctico del plan de trabajo diseñado por la dirección técnica de RTRM para incorporar la tecnología HbbTV en la dotación de contenidos interactivos a la emisión de 7RM.

5.1 Plan de desarrollo

Recordando el plan definido previamente en el prólogo:

- 1 – Estudiar la conveniencia de llevar a cabo desarrollos para la plataforma HbbTV.
- 2 – Participar en el grupo de radiodifusores para el estudio de la televisión interactiva.
- 3 – Realización de un piloto y lanzamiento de una aplicación básica.
- 4 – Lanzamiento de la aplicación definitiva.

Los puntos 1 y 2 de este plan han sido ya cubiertos en los capítulos 1 y 2 de este proyecto. Este capítulo 3 se corresponde con el desarrollo de las tareas 3 y parte de la tarea 4, como quedaba inicialmente definido en el prólogo.

5.2 Piloto

Al decidimos por la tecnología HbbTV creímos conveniente hacer un piloto que nos sirviese para saber que dificultades nos encontraríamos en la puesta en marcha, desarrollo y mantenimiento de nuestra futura plataforma. Si la experiencia resultaba satisfactoria (como finalmente sucedió) se exigiría el equipamiento soporte para HbbTV en el nuevo contrato de la red de transporte y difusión de 7RM. El nuevo contrato se inició el 1 de julio de 2012.

Para el piloto nos pusimos en contacto con la empresa Abertis ya que ésta nos ofreció su colaboración sin compromiso, como había realizado con otras televisiones. El trabajo se centró en base a tres puntos:

- 1 – Diseño de la apariencia de la aplicación para la fase de pruebas.
- 2 – Método para incorporar (ingesta) contenidos de video bajo demanda.
- 3 – Señalización.

Los resultados de este piloto nos servirían como base de la futura plataforma HbbTV.

5.2.1 Aplicación básica

Para la aplicación decidimos usar una versión básica que ofreciese algunos vídeos para ser descargados por streaming. Así que nos centramos en definir la apariencia debía tener para que fuese en línea con la imagen corporativa de 7RM. La navegación por los distintos apartados de la aplicación es también muy sencilla. Consiste en seleccionar uno de los video propuestos y luego reproducirlo desde la ventana del reproductor. Cuando el vídeo se está reproduciendo se puede pausar o posicionarse sobre la barra de desplazamiento del video para ir a otra posición en la reproducción. El reproductor de vídeo también puede guardar una cookie que recuerde la última posición en un video que no se ha terminado de ver. También tiene un botón "Home" arriba que te permite volver a la pantalla principal donde se ve el video broadcast en una ventana y se listan los vídeos on demand disponibles. En todo momento el botón rojo del mando finaliza la aplicación.

Esta es una captura de pantalla del aspecto final de la aplicación con video broadcast incrustado:



Figura 5.1 – Captura de la pantalla principal de la aplicación con un video broadcast.

En esta captura podemos ver la reproducción de un video broadband:



Figura 5.2 – Captura de pantalla durante la reproducción de un video broadband.

5.2.2 Ingesta de contenidos

La ingesta de contenidos consiste en codificar y dar formato a los vídeos para que sean incorporados a la base de datos. Luego el servidor de streaming los sirve a los clientes HbbTV que los soliciten. El streaming es HTTP tal cual se ha explicado anteriormente en este documento. El formato elegido para los vídeos es el especificado en el documento para SD y para HD. Es decir, un contenedor MP4 con codec de vídeo h.264 y audio AAC. Las resoluciones y bitrates son también las del documento que vienen especificadas en el estándar.

Al decidir el formato del contenedor, teníamos dos opciones MP4 o TS. Elegimos MP4 por varios motivos:

- Separación del contenido de los metadatos: En MP4 se pueden guardar aparte mientras que en TS va todo multiplexado. Podríamos tener un servidor de subtítulos y otro de audio/vídeo.
- Almacenamiento de pistas independiente: Al igual que antes, las esencias de vídeo y audio no van multiplexadas van en pistas independientes que pueden estar en el mismo contenedor o en contenedores independientes. Esto permitiría que hubiese un servidor con los vídeos y otro con los audios, y se podrían combinar para reproducción usando XMLs. También se podría tener un servidor que contuviese solo el audio relativo a la audiodescripción, de tal forma que cuando estuviese disponible se actualizase el XML y se ofreciese al cliente sin tener que volver a empaquetar el MP4.
- Trickplay avanzado: Se puede hacer trickplay más avanzados en base a los box *moof* y *mdat*.
- Retrocompatibilidad con TS: Si se guarda la información de temporizado del MP4 en un fichero distinto al del audio/vídeo se puede convertir un MP4 en un TS. Esto realmente es útil para servidores de vídeo por internet que tengan los contenidos en MP4 y los quieran servir a dispositivos Apple.
- DRM: El diseño del MP4 contemplaba la integración con un DRM y soporta CENC (Common Encryption). TS se puede proteger (Blu-Ray).
- DASH, Ultraviolet (multi DRM) apuestan por MP4 y soportan TS solo para dispositivos antiguos.

A todo esto añadimos que en las conversaciones que hemos tenido con los fabricantes, todos ellos nos han indicado que sus terminales solo iban a dar soporte a MP4 para el canal broadband. En el mundo del streaming web, solo Apple usa TS (M2TS) el resto (Microsoft, Adobe, Google (excepto para iPhone), usan MP4.

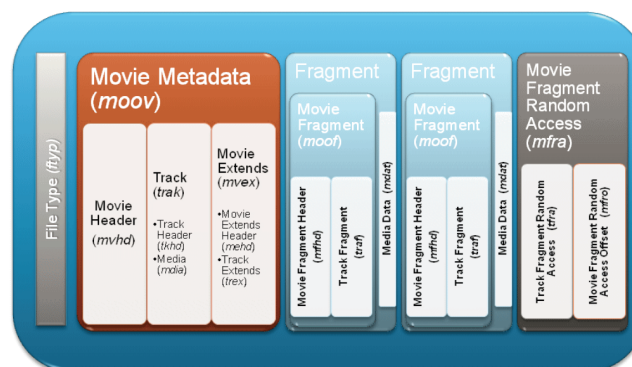


Figura 5.3 – Detalle de la estructura de un MP4 según Smooth Streaming de Microsoft .
<http://alexzambelli.com/>



Ya sabemos que vamos a codificar nuestros contenidos en un MP4 con h.264+AAC, pero no tenemos ningún procedimiento claro para hacerlo.

Debemos buscar una forma rápida y efectiva de hacerlo, pues son muchos contenidos los que hay que codificar y generalmente son de duraciones superiores a los 30 minutos. El principal problema en este punto es el origen de los contenidos. Los contenidos se recuperan del archivo audiovisual, que están en un formato profesional.

Las herramientas de codificación multimedia más habituales no están preparadas para trabajar con formatos profesionales como origen de codificación. Las que si lo están son herramientas profesionales que tienen un coste elevado.

El siguiente problema es dar el formato correcto al contenedor MP4 para que pueda hacerse el streaming progresivo o "pseudostreaming" por HTTP. Esto consiste en que el box *moov* (ver figura de la página anterior) debe estar colocado al inicio del archivo MP4. El *moov*, se puede ver como una especie de índice de tiempos del archivo. Si el *moov* es lo primero que se descarga, el cliente de reproducción sabrá como es el MP4 de antemano y podrá pedir fragmentos arbitrarios sin necesidad de descargarlos previamente.

Generalmente todos los programas de codificación colocan el *moov* al final por un motivo muy simple, y es que para poder poner el *moov* al principio hay que reservar un espacio y ese espacio va en función de la información del MP4 cuando está ya codificado, por tanto, hasta que el MP4 no esté codificado por completo no se sabe cuanto ocupa ni que tiene que llevar el *moov*, por eso se pone al final del proceso de codificación.

Por tanto debemos buscar una herramienta para codificar formatos profesionales a formatos no profesionales y que además sea capaz de preparar el MP4 para el pseudostreaming, que cabe recordar que es un requisito del estándar.

Tras hacer muchas pruebas, el esquema pasaba por codificar por un lado con un programa y mover el *moov* después con otro programa. Este esquema era funcional pero no se podía hacer todo el proceso de una vez de una forma sencilla.

Tras estudiar diferentes herramientas, encontramos una versión del conocido programa *ffmpeg* especial para broadcast llamada *ffmbc*, que nos permitía codificar en h.264+AAC en un MP4 y al finalizar el proceso de codificación dejaba el *moov* al principio del archivo. Como trabajamos en entorno OS X nos tocó compilarlo pues no hay un ejecutable precompilado para esta plataforma (si lo está para windows). Un ejemplo de línea de código para codificar un MP4 para streaming en el portal HbbTV:

```
ffmbc -i archivo.mxf -map_audio_channel 0:1:0:0:1:0 -map_audio_channel 0:2:0:0:1:1 -vcodec libx264 -b 1500k -aspect 16:9 -acodec libfaac -ar 48000 -ab 96k -ac 2 -vf "[in] yadif=0:1:0 [progresivo], [progresivo] scale=1080:810 [progresivoescalado];movie=/$HOME/Desktop/Logo7RM.png [logo]; [progresivoescalado][logo] overlay=W-w-50:50: [out]" -faststart "auto" archivoHbbTV.mp4
```

En este ejemplo, los diferentes parámetros y opciones elegidos nos permiten codificar, escalar, desentrelazar, mapear los canales de audio y poner un logo. La opción *-faststart "auto"* es la que mueve el *moov* al principio tras la codificación.



5.2.3 Señalización

Ahora que tenemos una aplicación y que sabemos como añadirle contenidos de video bajo demanda que sean compatibles con el estandar HbbTV tenemos que señalar la aplicación en la tabla AIT de los TS de la emisión TDT de 7RM y de 7RM HD.

Como es una aplicación que el receptor descargará exclusivamente por internet, lo que se señalará en la tabla AIT es la URL de la aplicación utilizando la etiqueta "*initial_path_bytes*" del descriptor "simple application location" de la tabla AIT.

Para esto había dos opciones, o poner un equipo completo que generase la tabla AIT y permitiese más cosas como generar y gestionar los datos del DSM-CC y sus TS correspondientes o poner simplemente un generador de tramas que repitiese una trama fija con una AIT señalizando la URL de la aplicación.

La opción del generador de tramas es válida por la simplicidad de la aplicación piloto. Al ser una aplicación completamente web por el canal broadband, no hay que monitorizar la AIT para cambios o actualizaciones del carrusel o stream events.

En este caso la AIT es simplemente un lanzador de la aplicación que no aportará nada más al ciclo de vida de la aplicación mientras el usuario esté en el mismo canal.

Para aplicaciones más complejas (veremos alguna más adelante en este apartado) un simple generador de tramas fijas no es válido ya que seguramente serían necesarias actualizaciones de la AIT, cambios en las aplicaciones disponibles según la programación, stream events, datos de carrusel, etc...

Esta fase de señalización se solapó con la entrada en vigor del nuevo contrato para la red de transporte y difusión TDT de 7RM, por lo que se pasó a usar el equipamiento definitivo propuesto por la empresa adjudicataria para prestar el servicio HbbTV.

5.2.4 Conclusiones fase piloto

Tras la fase piloto pudimos comprobar la potencia que nos ofrecía este estándar sobre todo con los contenidos ofrecidos por el canal de banda ancha. Con muy poco esfuerzo es posible tener una aplicación de video bajo demanda con contenidos y es fácil hacerlos llegar a los usuarios que tengan este tipo de dispositivos compatibles.

La inversión necesaria es bastante contenida y la posibilidad de reutilizar infraestructuras existentes para el desarrollo de aplicaciones es muy alto, pues para servir la aplicaciones se van a usar los mismos sistemas que para las aplicaciones web ya existentes.

Por tanto, la decisión de ofrecer contenidos interactivos usando el estándar HbbTV nos parece acertada vista la facilidad de desarrollo y los bajos costes de implementación.

En los apartados sucesivos vamos a perfilar cual va a ser el escenario definitivo y su arquitectura.



5.3 Arquitectura definitiva

La arquitectura del escenario final soporte de las aplicaciones HbbTV de 7RM se divide en tres ámbitos que son:

1 – La cabecera de red. Aquí es donde se ubican los equipos que van a señalar las aplicaciones mediante tablas AIT.

2 – Los servidores de la aplicación y sus servicios. Que ya está desarrollado pues es la infraestructura que usamos para distribuir los contenidos de las páginas web para ordenadores personales y dispositivos móviles.

3 – La ubicación del terminal. Generalmente en la casa del usuario.

Vamos a detallar un poco las dos primeras, pues la tercera consiste en el navegador HbbTV que tenga instalado el terminal (televisor, STB, PVR, etc...)

5.3.1 Cabecera de red

Como hemos comentado anteriormente, el 1 de julio de 2012 se inició el contrato para el servicio de transporte y difusión de la señal TDT de 7RM en el que se incluía soporte HbbTV.

La empresa adjudicataria ha instalado una plataforma denominada ISIS-HbbTV que tiene las siguientes funciones:

En cuanto a señalización:

- Señalización de las aplicaciones HbbTV en la tabla AIT.
- Inserción de la tabla AIT en la señal multiplexada TDT.
- Monitorización y mantenimiento de la plataforma.

Además, la misma plataforma ofrece los siguientes servicios:

- Sistema de ingesta de contenidos.
- Almacenamiento de contenidos.
- Aplicación HbbTV de acceso a contenidos de video.
- Utilización de la plataforma de vídeo bajo demanda.
- Tráfico CDN.

Esta parte de la plataforma es la misma que hemos estado utilizando para el piloto, con lo que en el mismo momento en el que entró en vigor la nueva contratación ya disponíamos de un desarrollo plenamente funcional. La aplicación piloto es algo limitada en cuanto a funciones y tenemos previsto hacer un desarrollo propio que disponga de más opciones, lo veremos más adelante.

El siguiente gráfico muestra como será la arquitectura a nivel HbbTV en la cabecera:

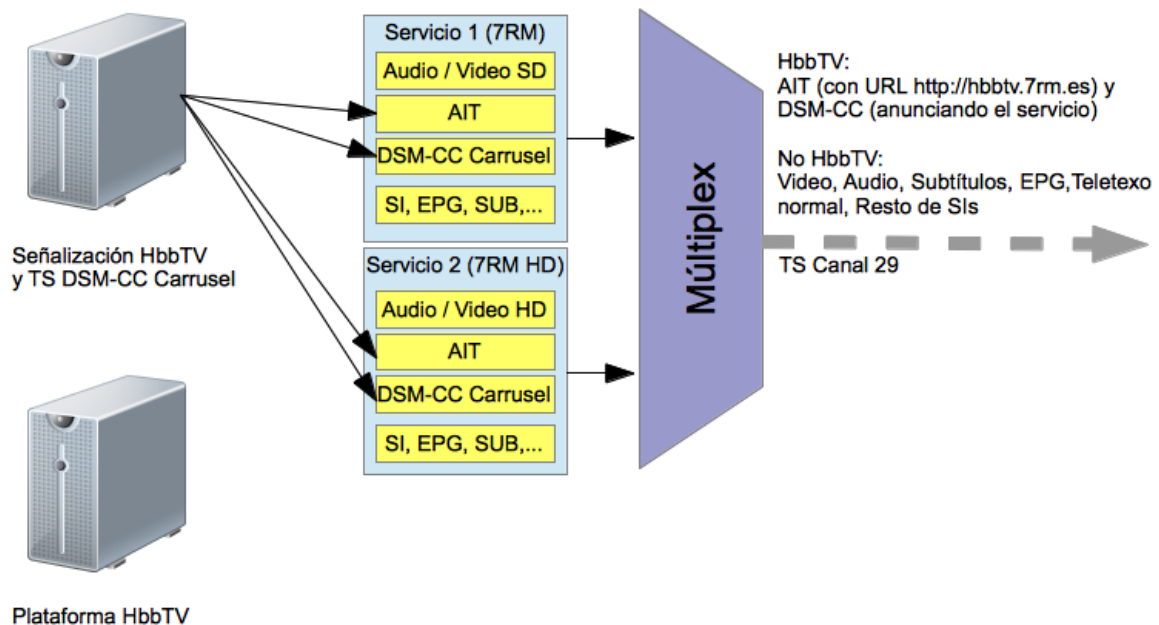


Figura 5.4 – Arquitectura de señalización en cabecera de red.

Por un lado vemos el equipo que señala y actualiza carruseles y por el otro el equipo encargado de servir la aplicación HbbTV y hacer la ingesta de contenidos. El tráfico CDN de este servidor lo ofrece la empresa que hace el hospedaje del equipo. Es decir, es la empresa que hospeda este servidor la que se encarga de la distribución de contenidos, utilizando para ello los equipos que considere necesarios para ofrecer una calidad de servicio mínima y una alta disponibilidad.

Con respecto al equipo encargado de la señalización, podemos ver la información que añade a cada servicio (canal). Concretamente introduce la AIT con la URL de la aplicación y un carrusel de datos que servirán únicamente para anunciar el servicio a los terminales que no dispongan de conexión a la banda ancha. No tenemos previsto utilizar el carrusel de datos para mucho más.

Así será la **primera fase** de implementación de HbbTV.

En la **segunda fase**, el equipo que proporcionará la aplicación y la ingesta, control y distribución de contenidos no va a ser la plataforma HbbTV de la imagen. La nueva aplicación que estamos desarrollando y que presentamos más adelante en este documento estará alojada en la misma ubicación que el actual portal web para ordenadores personales y dispositivos móviles. La distribución de contenidos también se hará desde esta ubicación. Nuestro proveedor de servicios en este caso es la Fundación Integra.

Hablamos de la distribución de contenidos en el siguiente punto.



5.3.2 Distribución de contenidos

La Fundación Integra (FI) nos proporciona servicios de soporte, distribución y mantenimiento de nuestras páginas Web para ordenadores personales y dispositivos móviles. En la primera fase de implantación del HbbTV, no vamos a usar los servicios de FI para el desarrollo o distribución de contenidos de la aplicación, pero si lo haremos en la segunda fase migrando la aplicación HbbTV y sus servicios a la plataforma que tenemos actualmente con ellos.

Detallamos a continuación la arquitectura del servicio y como incorporaremos la aplicación HbbTV.

Desde que iniciamos el desarrollo de las páginas web *www.7rm.es* para ordenadores personales y para móviles con sus servicios de streaming asociados se tuvo siempre en mente que todas ellas pudiesen ser actualizadas a la vez y que el sistema para introducir la información fuese sencillo y común a todas. Lo que pretendíamos era que, por ejemplo, al subir una noticia, esta se publicara automáticamente en todas nuestras plataformas que mostrasen noticias (teletexto, web, aplicaciones para móviles y tablets, aplicaciones interactivas, etc...). Lo mismo aplica para un video de la sección "A la carta". Para ello hay un gestor de contenidos que ofrece una interfaz para crear, actualizar y subir nuevos contenidos que luego serán visibles en la web. Este gestor de contenidos es accesible por un interfaz (en Java), que es el que usan las aplicaciones web para presentar y recuperar los contenidos del gestor. Todo esto se hace siguiendo el modelo vista-controlador.

En esta segunda fase enlazaremos la aplicación HbbTV con el gestor de contenidos para que se presenten los contenidos que consideremos convenientes en la aplicación HbbTV. Uno de los contenidos que queremos presentar en la aplicación HbbTV serán los vídeos de la sección "A la carta" con sus correspondientes secciones y los informativos. Otros contenidos que queremos incorporar a la aplicación HbbTV será la EPG (información de la programación de los próximos días), el tráfico, agenda cultural, etc. de forma similar a como hicimos para las aplicaciones para móviles (iOS y Android).

Para el video "A la carta", tenemos que solucionar previamente un problema. Desde que están funcionando las distintas páginas web, la codificación de los vídeos bajo demanda (sección "A la carta") se ha hecho en MP4 con h.264 para el video y AAC para el audio, lo que es correcto para usar en HbbTV. El streaming para las web se hace usando FMS (Flash Media Server) y Wowza para móviles.

El problema que hay que solventar es que durante todo este tiempo al generar el MP4 el *mooV* se ha quedado al final del archivo, y esto no cumple con el estándar HbbTV. Esto no ha impedido que haya funcionado el streaming progresivo para la web y para los móviles ya que desde hace un tiempo, los servidores Web llevan incorporado un módulo para hacer streaming progresivo de archivos aunque no tengan el *mooV* al principio. Se suele llamar streaming progresivo pero realmente lo correcto es "pseudostreaming". Estos módulos funcionan permitiendo que el reproductor de la web se posicione sobre cualquier punto del archivo, y lo que hace es traducir una posición temporal en la barra del reproductor a una posición en bytes del archivo MP4.

Para solventar el problema del *mooV* usaremos un programa que individualmente irá procesando todos los archivos MP4 y colocando el *mooV* al principio del archivo. Se hará en tandas y primero haremos pruebas para comprobar que no afecta al resto de servicios ya que lo que se tocará será el MP4 maestro.

En el siguiente gráfico podemos ver en un esquema lo que acabamos de comentar:

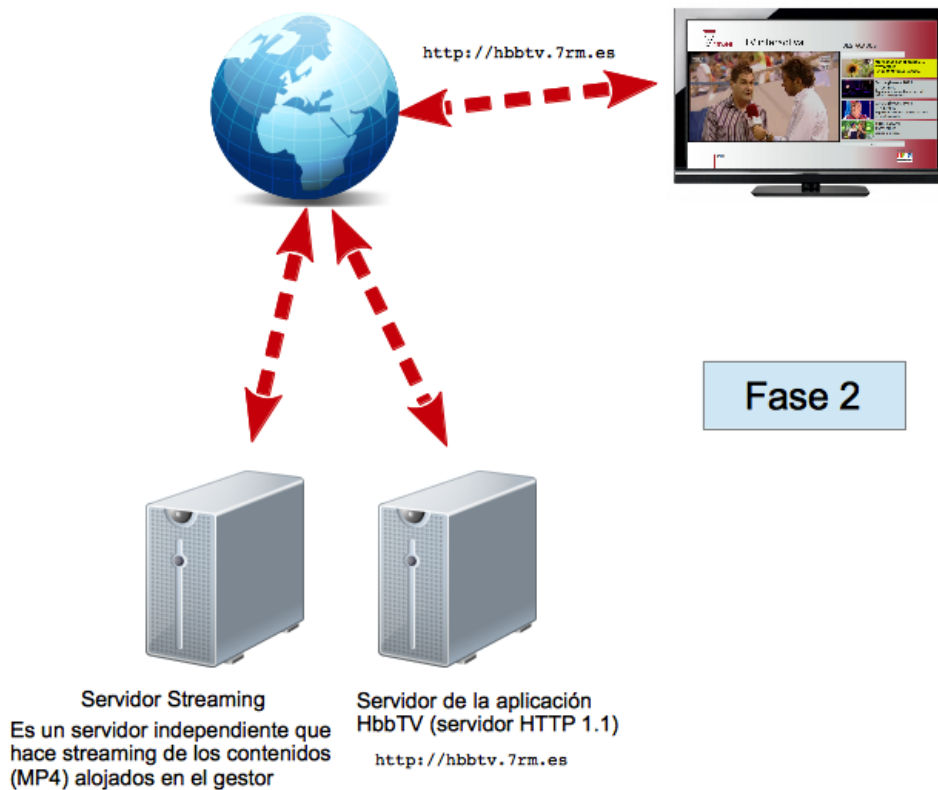


Figura 5.5 – Arquitectura en Fase 2 de distribución de la aplicación y de los contenidos on demand.

Este esquema se corresponde con el escenario final. Un servidor se encargará de la distribución de la aplicación a los terminales HbbTV y el otro se encargará exclusivamente de hacer el streaming del video bajo demanda de los contenidos “A la carta”. Internamente ambos estarán conectados al gestor de contenidos (que no aparece en el gráfico).

5.3.3 Terminal HbbTV del usuario

Este es el tercer ámbito de ejecución de las aplicaciones HbbTV. Es el único punto de la cadena no controlado por el radiodifusor. Lo que nos interesa del terminal son las funciones y la implementación del estándar que haga el navegador del terminal, que es quien ejecuta las aplicaciones.

La pila de tecnologías que usa HbbTV está funcionando en mayor o menor medida en los navegadores de todas las SmartTV que hay en el mercado para sus portales de marca o en los navegadores que incorporan para la navegación por internet. Estos navegadores no son navegadores PC adaptados a la TV, son navegadores diseñados específicamente para la TV por empresas especializadas. Como vamos a ver más adelante, estos navegadores incluyen soporte para las tecnologías web de la TV interactiva sobre los que se construye HbbTV aunque en algunos casos no incluyen el perfil específico de HbbTV.



Comercialmente HbbTV aún no está siendo ampliamente explotado y mucha gente no sabe que tiene un terminal que es compatible. Es más, hay algunos modelos de terminal que en la configuración inicial, si se les indica que el país es Francia o Alemania, activan la función HbbTV y si se les indica cualquier otro país no se activa.

En las conversaciones que hemos mantenido con fabricantes, nos trasladan que hay un parque bastante extenso de terminales SmartTV con el HbbTV “dormido” pero que en cuanto el terminal se conecte a internet es posible “activarlo”, hay otros fabricantes que dicen que el soporte está, pero no es posible activarlo.

Para ilustrar como se implementan las tecnologías en un navegador de TV vamos a poner un par de gráficos con los diagramas en bloques:

A) TARA Systems

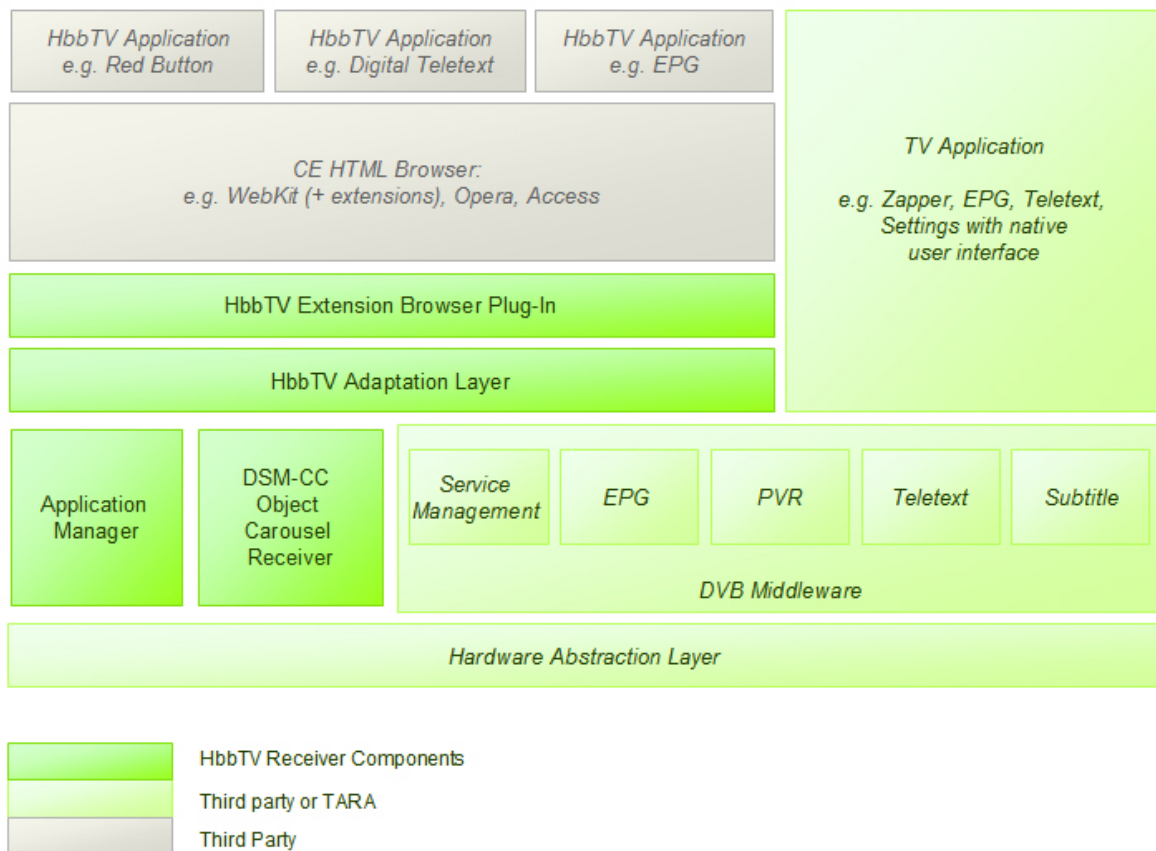


Figura 5.6 – Arquitectura de la solución propuesta por Tara Systems.

<http://www.tara-systems.de/hbbtv.html>

En este caso, la empresa TARA Systems, propone una serie de módulos interoperables para dar soporte a los requisitos HbbTV. Vemos un módulo para el “Application Manager”, otro que gestiona los carruseles DSM-CC, una capa de adaptación de HbbTV y otra que proporciona extensiones HbbTV al navegador, todos ellos en verde oscuro. Lo que hay en verde claro son las aplicaciones típicas locales de la televisión (subtítulos, EPG del fabricante, configuraciones,...) que TARA dice que pueden ser suyos o de un tercero.

Lo que hay en gris realmente es un navegador CE-HTML que puede ser el motor WebKit, el navegador Opera, Chrome, etc... al cual se le incorporarían las funciones HbbTV con los módulos de color verde oscuro. Sobre el navegador que el fabricante haya decidido poner se ejecutan las aplicaciones HbbTV. En definitiva lo que se propone aquí es un sistema operativo para el televisor que incluye las funciones típicas del terminal, junto con módulos específicos HbbTV que se enlazan con el navegador

B) ACCESS

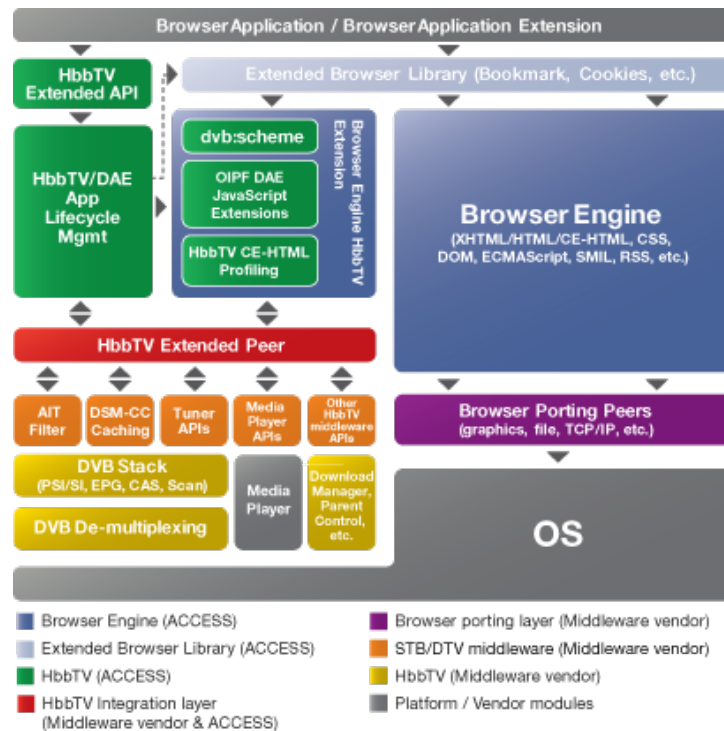


Figura 5.7 – Arquitectura de la solución propuesta por ACCESS.

<http://www.access-company.com/>

A diferencia del caso anterior, la empresa ACCESS ofrece su propio navegador (módulo de color azul), con extensiones HbbTV (verde) y extensiones no HbbTV (azul claro). El navegador soporta CE-HTML, CSS, DOM, JavaScript (ECMAScript), entre otros. El gráfico reproduce el resto del entorno que rodearía a estos componentes como son el SO del terminal, media player, componentes que leen la señal DVB (TS, tablas SI, ...).

Los bloques en color naranja, realizan las funciones de filtrado de la AIT, lectura de los carruseles DSM-CC, proporcionan un API para manejar el sintonizador, el reproductor multimedia, y otras funciones HbbTV que podrían ser las opcionales como las de descarga o el PVR. Son funciones implementadas mediante middleware y no se han incrustado dentro del navegador. Exceptuando la última, el resto no son específicas de HbbTV y podrían ser válidas para MHP, por ejemplo. Incorporarlas como middleware implica que un tercero tiene que hacer software para realizar esas funciones, pero simplifica la tarea del desarrollador del navegador ya que le permite centrarse exclusivamente en el navegador. Dicho de otro modo, el middleware de la AIT ya le daría el AIT filtrado a los módulos HbbTV, el desarrollador del navegador no tiene que preocuparse de como filtrarlo solo hacer una llamada al middleware que le devolverá los datos de la AIT.

Este es el navegador con el que está equipado nuestro terminal de pruebas.



5.4 Desarrollo de la aplicación definitiva

Para poner en marcha el sistema, hemos estimado que era suficiente con una aplicación sencilla que ofreciese algunos vídeos bajo demanda de nuestra sección “A la carta” a la par que incrustaba la señal broadcast en la aplicación. En el proceso de puesta en marcha, que ha durado varios meses, hemos visto los cambios que tenemos que hacer a nuestra infraestructura actual, hemos hecho muchas pruebas para encontrar flujos de trabajo compatibles con HbbTV y hemos visto que queremos reforzar.

También hemos visto decenas de aplicaciones de otros radiodifusores nacionales y europeos. Hemos hecho diversas valoraciones y hemos establecido un criterio acerca de cuales son los servicios que queremos que estén disponibles en nuestra aplicación final.

Al ser una televisión pública, nuestro objetivo es de servicio público al ciudadano.

Ofreceremos dos tipos de servicio, por un lado, contenidos de video on demand fundamentalmente de producción propia y enfocados a la difusión de las señas de identidad de la Región de Murcia (programación “A la carta”, informativos) y por otro aplicaciones interactivas con información actualizada como teletexto digital avanzado, EPG, agenda cultural de la Región, el tiempo, tráfico en carreteras, avisos del centro de emergencias 112, etc... Además incorporaremos aplicaciones web para servicios de la administración regional como la “Cita médica” del Servicio Murciano de Salud (en su momento se lanzó en MHP) buscando favorecer su accesibilidad trasladándolo al entorno del televisor, que aunque técnicamente es más limitado que el ordenador, está más extendido y es más fácil de utilizar.

5.4.1 Consideraciones en cuanto al diseño de la aplicación

Antes de decidir que contenidos concretos vamos a incluir en la aplicación final, debemos ver cual es la mejor manera de plantear la aplicación buscando que sea fácil de utilizar y que el usuario se encuentre cómodo usándola.

Este es un punto vital para la supervivencia de la aplicación y del servicio, ya que si hacemos virguerías tecnológicas que luego generen rechazo en los usuarios por un diseño poco cuidado o confuso, el servicio no tendrá ningún futuro.

También hay que cuidar que aplicaciones se incluyen porque si el usuario no ve nada útil cuando entra a la aplicación, por muy bien diseñada que esté no encontrará ningún aliciente para seguir usándola. Para asistir en la creación de aplicaciones interactivas, el propio estándar tiene un punto informativo en el que da unas directrices de implementación.

Los conceptos que se tratan en este punto, y que deberían ser tenidos en cuenta al tomar las decisiones en cuanto a diseño y funcionalidad son los siguientes:

- Balance entre video y aplicación.
- Si se quieren aplicaciones estáticas o que cambien con el tiempo (o ambas).
- Que botones queremos que sean necesarios para controlar la aplicación.
- Tipos de acceso a la aplicación interactiva (Botón rojo, botón TXT, enlace durante un programa, ...).
- Como salir de la aplicación (Exit, cambio de canal,...) y/o volver al menú principal.



Además de esto, es muy importante tener en cuenta el conocido como “10 foot user interface”. Este concepto hace referencia a los interfaces gráficos de usuario (GUI) que están pensados para ser presentados en una pantalla de TV que está a una cierta distancia (10 foot = 304,8 cm) del usuario y que son controlados con los (escasos) botones de un mando a distancia.

Son los típicos interfaces que se encuentran en los reproductores multimedia. La navegación por los menús es muy rápida, se puede subir y bajar niveles con facilidad y el tamaño de los iconos es suficientemente grande como para que de un solo vistazo se sepa cual es su función.

Un ejemplo de un diseño 10 foot UI:

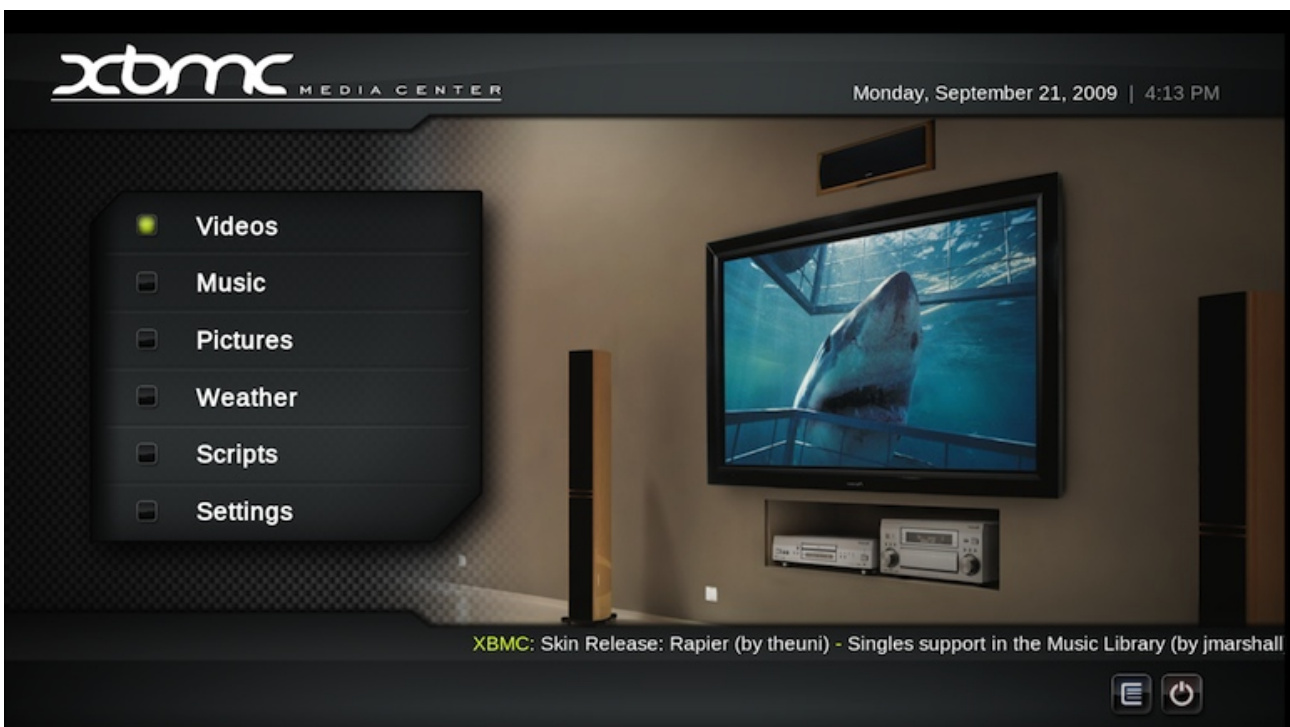


Figura 5.8 – Captura de pantalla de un interfaz “10 foot UI”.
<http://upload.wikimedia.org/wikipedia/commons/8/8f/Screenshot000.png>

Esto es una captura de pantalla de una distribución de Linux llamada XBMC. Es un media center gratuito open source. El diseño de los menús es de un tamaño suficiente como para que se lean perfectamente todas las opciones del menú a tres metros de distancia. La navegación entre ellos también está muy bien cuidada y es simple ya que con pocas pulsaciones de las teclas del mando se llega a todas las opciones.

Algunas de las aplicaciones que hemos visto de otros radiodifusores, aunque son muy buenas, nos han parecido muy sobrecargadas de opciones y contenido en muy poco espacio. Recordemos que las aplicaciones HbbTV tienen asignada una resolución de 1280x720.

Creemos que si tenemos en cuenta las directrices de diseño para aplicaciones interactivas y las diseñamos según un estilo “10 foot UI” aumentará la usabilidad de la aplicación, que al final es de lo que se trata.



En cuanto a la navegación del usuario por la aplicación queremos añadir las conclusiones de France Télévisions tras su desarrollo piloto de una aplicación para Roland Garros que finalmente estrenarían en el torneo de 2012.

Reproducimos el original:

“With hindsight, the most complicated part, which must in no way be ignored, in the design of an interactive TV application, is the ergonomics of the navigation process. In our initial design, we had already identified this factor and considered the various solutions offered. However, after the two week Roland Garros trial, and with tests by numerous inexperienced users in our studios, there was no denying that navigational ergonomics of a service using only a TV remote control system was a crucial element on which the success of a service would be directly dependent.”

que viene a decir,

“echando la vista atrás, la parte más compleja, que no debe dejarse de lado bajo ningún concepto en el diseño de una aplicación de TV interactiva, es la ergonomía del proceso de navegación. En nuestro diseño inicial, lo tuvimos en cuenta y valoramos distintas soluciones. Sin embargo, tras las dos semanas de pruebas con la aplicación de Roland Garros, y con los test hechos en nuestros estudios por muchos usuarios inexpertos, no hubo duda de que la ergonomía de navegación de un servicio usando solo el mando a distancia era un elemento crucial del que dependía directamente el éxito del servicio.”

Bernard Fontaine
Director Técnico de nuevos desarrollos
Revista EBU tech-i de Junio de 2010

5.4.2 Entorno de desarrollo de las aplicaciones

Para desarrollar aplicaciones HbbTV no es necesario ningún programa ni sistema en especial. No hay herramientas específicas. Lo único específico que hay que conocer es el lenguaje CE-HTML que ha sido adoptado por HbbTV y el BAE (Browser Application Environment) del que hablábamos en la Parte 2 de este documento. El resto de conocimientos que hacen falta son acerca de tecnologías web estándares como JavaScript, CSS y XHTML.

Solo harán falta herramientas específicas si vamos a usar el canal broadcast con el protocolo DSM-CC para distribuir contenidos. No es la intención a largo plazo, pero más adelante intentaremos utilizar el canal broadcast para distribuir stream events.

En un principio si que tenemos la intención de distribuir un anuncio del servicio HbbTV por el canal broadcast para televisores que, siendo compatibles, no estén conectados al canal broadband. En el anuncio informaríamos de las ventajas que hay si se conecta el televisor a internet.

Para probar las aplicaciones ni siquiera hace falta un terminal real. Existen aplicaciones que simulan ser un terminal HbbTV y que permiten ejecutar aplicaciones. Concretamente para esto hay dos opciones, una sería el navegador Firefox con el complemento FireHbbTV. La otra opción es una pequeña distribución Linux que la compañía Opera ha desarrollado como entorno de desarrollo y prueba de las aplicaciones.

El complemento FireHbbTV es más flexible, pues está concebido como una ayuda al desarrollo de aplicaciones que, por ejecutarse en Firefox, también puede usar los cientos de complementos que éste tiene para el desarrollo de aplicaciones.

Además al ejecutarse en el entorno normal del sistema operativo se pueden seguir usando las



herramientas de preferencia de cada uno. Funciona inyectando código del API HbbTV al vuelo sobre el navegador Firefox. Las funciones que ofrece este complemento son:

- Detectar cuando una página es HbbTV por los encabezados presentes en el documento. Detecta encabezados `application/vnd.hbbtv.xhtml+xml` y `application/xhtml+xml`.
- Permite simular el mando a distancia con el teclado.
- Escala proporcionalmente el documento HbbTV si cambia el tamaño de la ventana de Firefox y te indica el factor de escalado.
- Muestra el área segura.
- Permite simular Stream Events DSM-CC. El objeto video/broadcast permite usar el método `addStreamEventListener()` para escuchar stream events.
- Hay algo de soporte para el protocolo `dvb://`. Se puede usar de tres formas:
 - 1) Abrir un documento local como si se hubiese descargado de un carrusel
 - 2) Pidiendo datos de un carrusel con `XMLHttpRequest`
 - 3) Con elementos de un XML AIT

El SDK de Opera también está concebido como un entorno de desarrollo pero de forma distinta. Es una imagen de máquina virtual de VirtualBox con un entorno de Linux en el que el entorno de desarrollo es el propio navegador Opera con su complemento DragonFly. Aunque sea una máquina virtual es un entorno cerrado donde solo se puede usar lo preinstalado por Opera.

Estas dos opciones son buenas para ayudar al desarrollo pero no se podrá terminar una aplicación y lanzarla al mercado sin antes haberla probado en un dispositivo real. Además habría que intentar probarlo en el máximo número de dispositivos posibles (TVs, STB, PVRs...) para asegurar que nuestra aplicación es lo más compatible posible. Por las conversaciones que hemos tenido con otros radiodifusores todos hemos tenido las mismas experiencias con las pruebas en dispositivos. Lo que en uno funciona bien en otro lo hace de manera distinta, más lento o directamente no funcionan algunas cosas. Por tanto es importante hacer pruebas en el máximo número de dispositivos e intentar perfilar o, llegado el caso, quitar de la aplicación las funciones que se puedan considerar más conflictivas.

También hay un proyecto francés denominado OpenHbb (www.openhbb.eu) que ha creado un entorno de desarrollo y prueba para aplicaciones HbbTV con el apoyo de las empresas más importantes del sector en Francia. Ese entorno iba a ser libre y abierto pero a día de hoy no hay nada que haya sido puesto a disposición del público.

Por tanto, como entorno de desarrollo para la aplicación utilizaremos editores de HTML, JavaScript y CSS básicos para escribir el código. Para probarlo en las primeras fases usaremos FireHbbTV y los ajustes finales serán en un terminal real.

Las aplicaciones que usaremos:

- Para escribir el código → Notepad++ (PC), Aptana y Amaya (PC, OS X) y Textwrangler (OS X).
- Para probar el código sin un terminal → Firefox + FireHbbTV (PC, OS X)
- Para probar el código con un terminal → VirtualBox con Ubuntu Server haciendo de servidor HTTP 1.1 (LAMP) y un terminal Engel EN2000

5.4.3 Presentación de la aplicación final

Para nuestra aplicación final vamos a optar por un formato en el cual, tras el gancho que invita al usuario a utilizar los servicios interactivos, saldrá una barra translúcida en la parte de abajo de la pantalla, dejando el vídeo broadcast en el fondo a pantalla completa.

Este es el esquema en bloques de lo que será dicha aplicación con sus funciones:

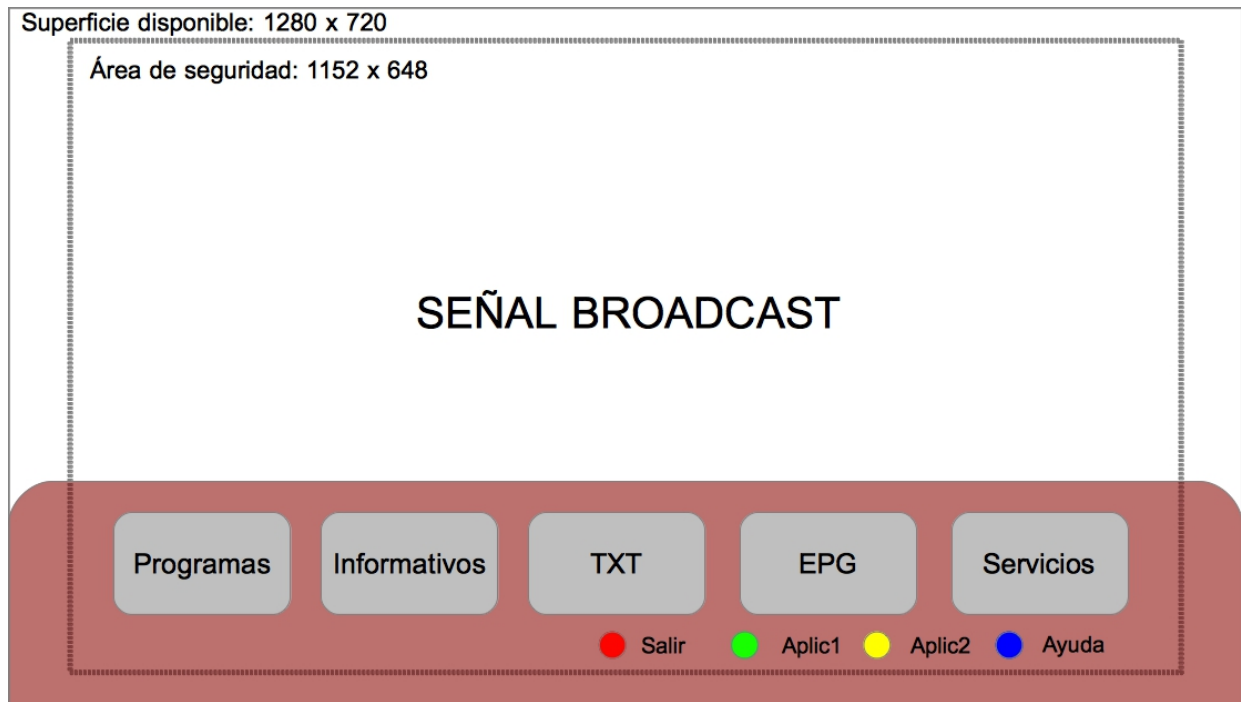


Figura 5.9 – Esquema en bloques de la aplicación definitiva.

El área de seguridad indica la zona límite a partir de la cual no deberíamos colocar contenidos que consideremos importantes ya que en cada televisor varía la superficie de visualización y si se ponen contenidos muy cerca del total disponible, esos contenidos podrían quedar fuera de la imagen y no verse. Por tanto, hay que respetar los márgenes de seguridad.

Las funcionalidades que tendrá cada opción serán:

PROGRAMAS: Se mostrará un lista de todos los programas que hay disponibles en la sección a la carta. El usuario seleccionará uno y se le mostrarán todos los vídeos disponibles.

INFORMATIVOS: Se mostrará un calendario para que el usuario seleccione la fecha del informativo y en esa fecha se le dará a elegir si quiere ver el informativo/deportes de la mañana, tarde o noche.

TXT: Se enlazará con la aplicación del teletexto digital.

EPG: Se mostrará la Guía Electrónica de Programación.

SERVICIOS: En este apartado se ofrecerán el resto de servicios de carácter público.

En el apartado “SERVICIOS”, la pantalla ofrecerá al usuario los servicios de carácter público que mencionábamos en el punto 5.4.

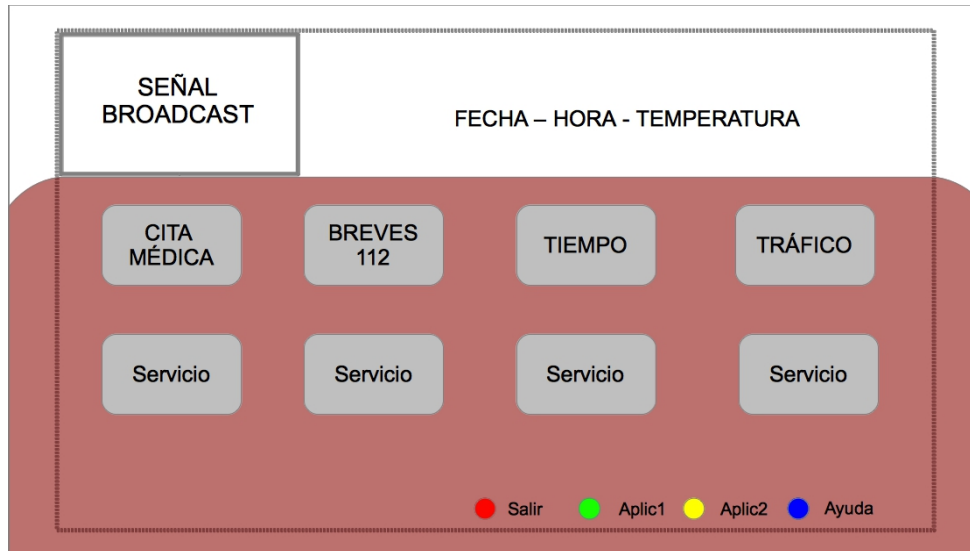


Figura 5.10 – Esquema en bloques de la aplicación definitiva en el apartado “SERVICIOS”

Aquí incorporaremos los servicios de Cita Médica que ya desarrollamos para MHP, noticias breves del 112, el tiempo, el tráfico de la DGT para nuestra comunidad y las aplicaciones que vayan surgiendo que resulten interesantes para el ciudadano.

En el apartado “INFORMATIVOS”, ofreceremos la posibilidad de elegir un día y ver cualquiera de los informativos de mañana, tarde o noche. Se elegirá día, se seleccionará entre informativos o deportes y luego se seleccionará si mañana, tarde o noche. Cuando se haya seleccionado todo, se pasará a reproducir el contenido. Al igual que en la aplicación piloto, se podrá avanzar el vídeo arbitrariamente y se podrá poner a pantalla completa.

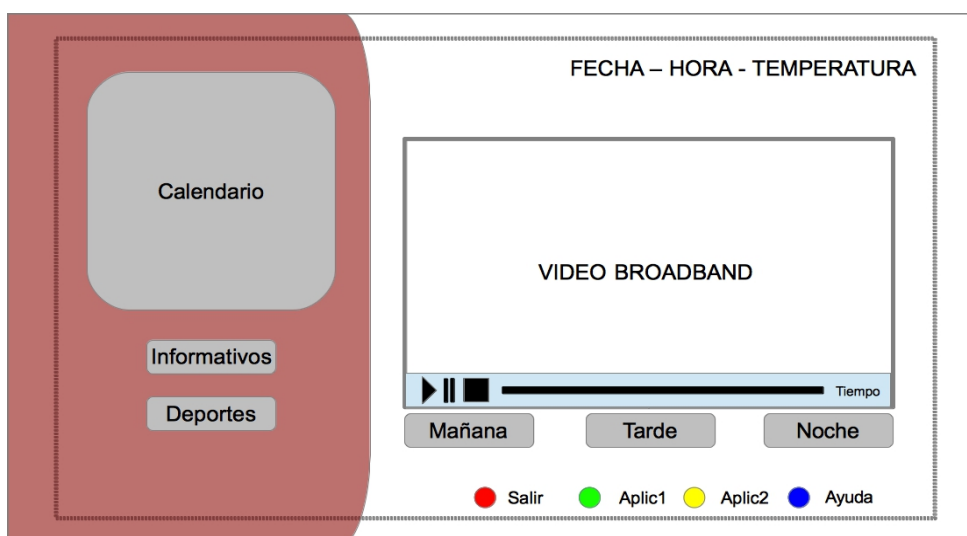


Figura 5.11 – Esquema en bloques de la aplicación definitiva en el apartado “INFORMATIVOS”

En el apartado “A LA CARTA” se mostrarán una selección de los programas con más visitas y al resto se podrá acceder navegando.

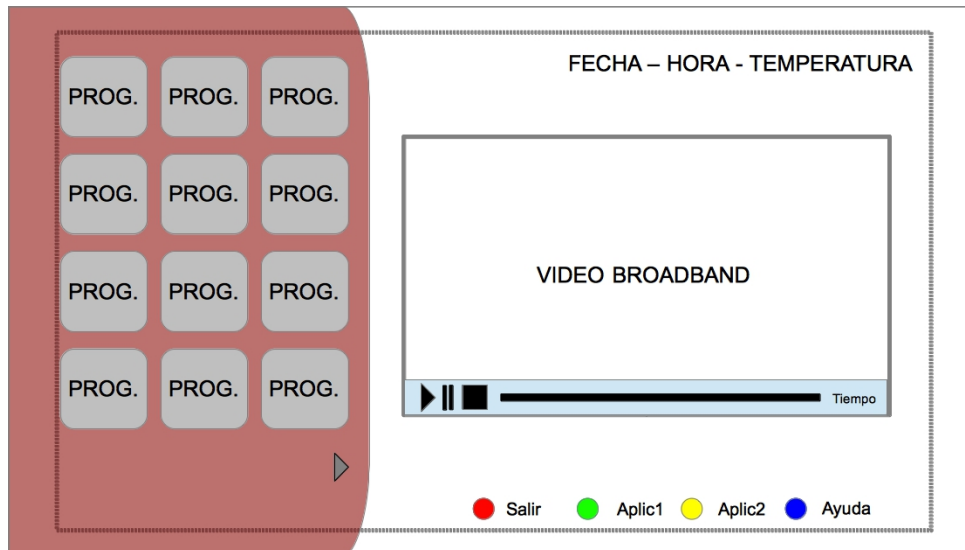


Figura 5.12 – Esquema en bloques de la aplicación definitiva en el apartado “A LA CARTA”

Al seleccionar el programa, se abrirá una lista ordenada cronológicamente para que el espectador busque el que más le interesa. Se indicará la fecha y el título del programa.

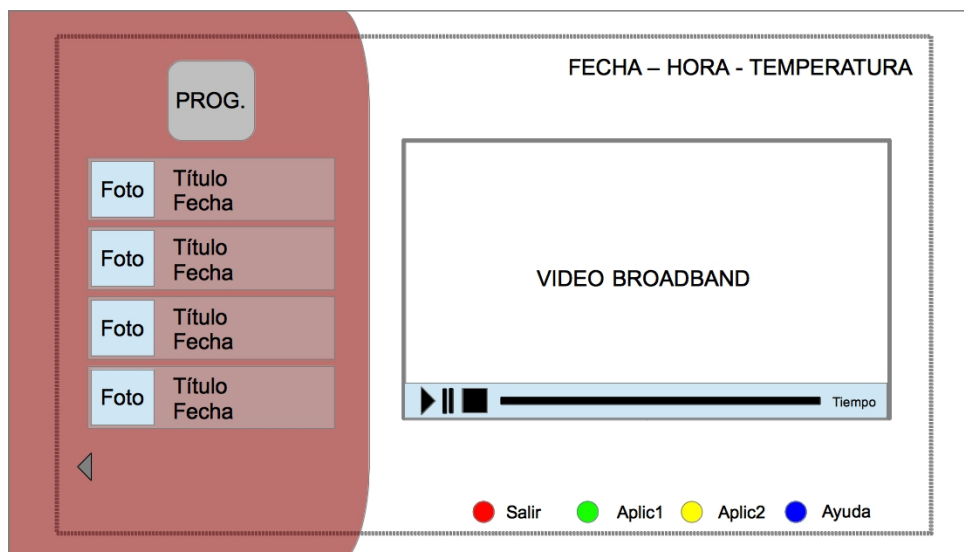


Figura 5.13 – Esquema en bloques de la aplicación definitiva en el apartado “A LA CARTA”

Los botones de colores se muestran en el esquema pero no tienen por qué estar presentes. De estarlo será para dar acceso directo a alguna función o para saltar a una aplicación tipo broadcast independiente (página web 7RM, por ejemplo). En el diseño de la aplicación se ha buscado que la navegación sea muy directa, que las aplicaciones sean fáciles de utilizar y que cualquier persona pueda encontrar lo que busca con facilidad.

5.5 Ejemplos de otros desarrollos y sistemas

Para finalizar vamos a exponer un ejemplo de un gran desarrollo HbbTV que ha tenido una gran repercusión en los medios. Se trata de la aplicación HbbTV que France Télévisions ha desarrollado para complementar las emisiones de torneo de Roland Garros de este año 2012. El año pasado ya hicieron un piloto, pero ha sido este año cuando lo han lanzado a todo el público.

A continuación el esquema técnico:

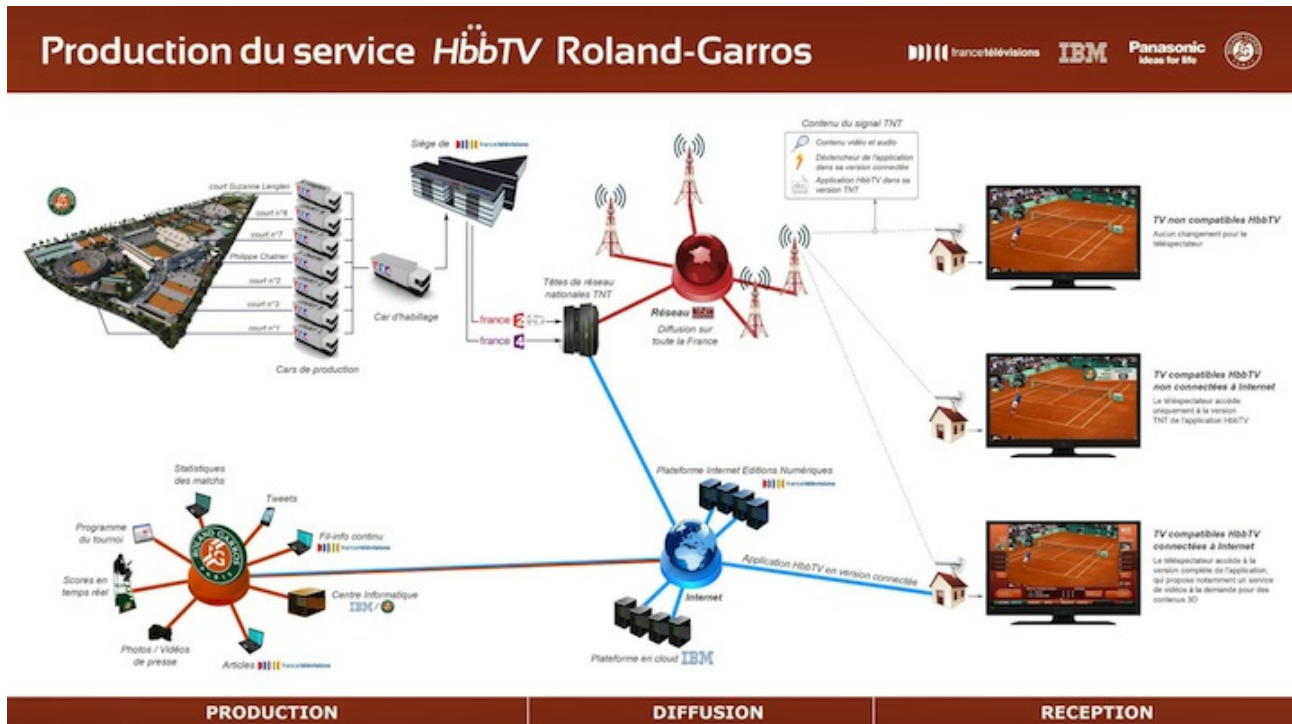


Figura 5.14 – Arquitectura del servicio HbbTV para Roland Garros desarrollado por France Télévisions.

Y una captura de pantalla de la aplicación en funcionamiento:



Figura 5.15 – Captura de pantalla de la aplicación HbbTV de Roland Garros.



Este esquema está dividido en tres apartados:

- Producción: Se observa como hay 7 unidades móviles que envían la señal A/V lineal a la sede de France Télévisions para los canales France 2 en HD y France 4. (El car d'habillage son los vestuarios y camerinos). Además en la parte de producción observamos como también hay contribuciones de datos para la aplicación HbbTV, es decir, no solo hay una contribución de las señales de audio/video lineales de los partidos de tenis, también hay contribuciones de datos para la aplicación HbbTV. Concretamente las contribuciones de datos consisten en artículos de los partidos, fotos y vídeos de prensa, Twitter, estadísticas...

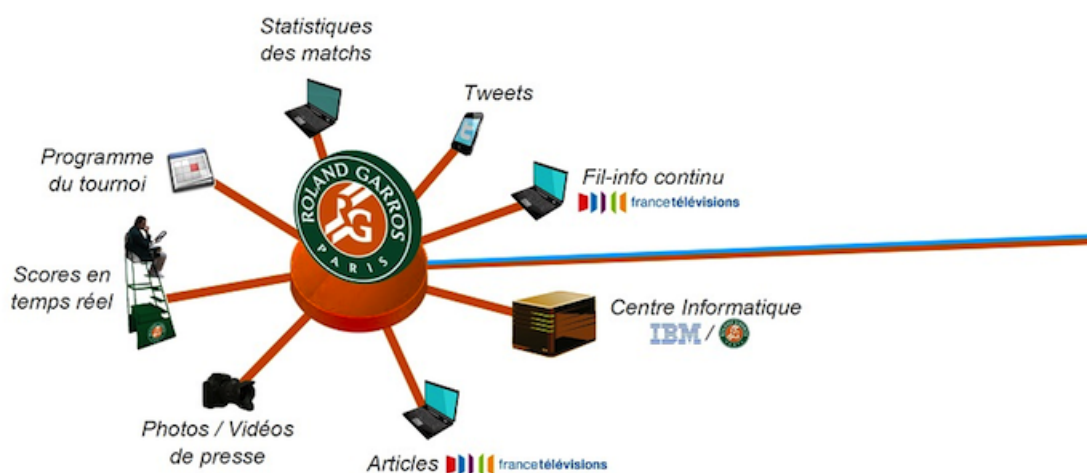
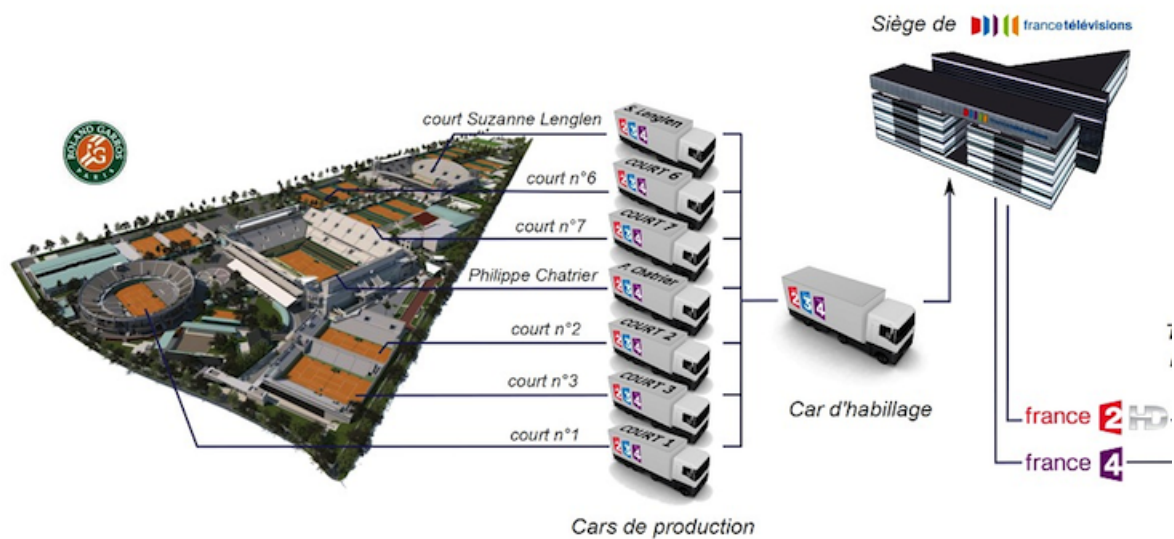


Figura 5.16 – HbbTV para Roland Garros. Detalle de el apartado de Producción.

- Difusión: De la sede se sube la señal a la cabecera de red y de ahí se hace la difusión para todo el territorio francés (línea roja). Por otro lado tenemos una infraestructura informática que ofrece un servicio de gestión de contenidos para la producción de datos y otra de distribución de contenidos en la nube. Esta infraestructura la monta IBM y está ubicada físicamente en Estados Unidos (al menos en el piloto del año pasado era así). Esta infraestructura también hace una contribución de datos a la cabecera de red, muy probablemente para los carruseles DSM-CC y señalización de stream events que seguramente son lanzados desde la contribución de datos de la parte de producción.

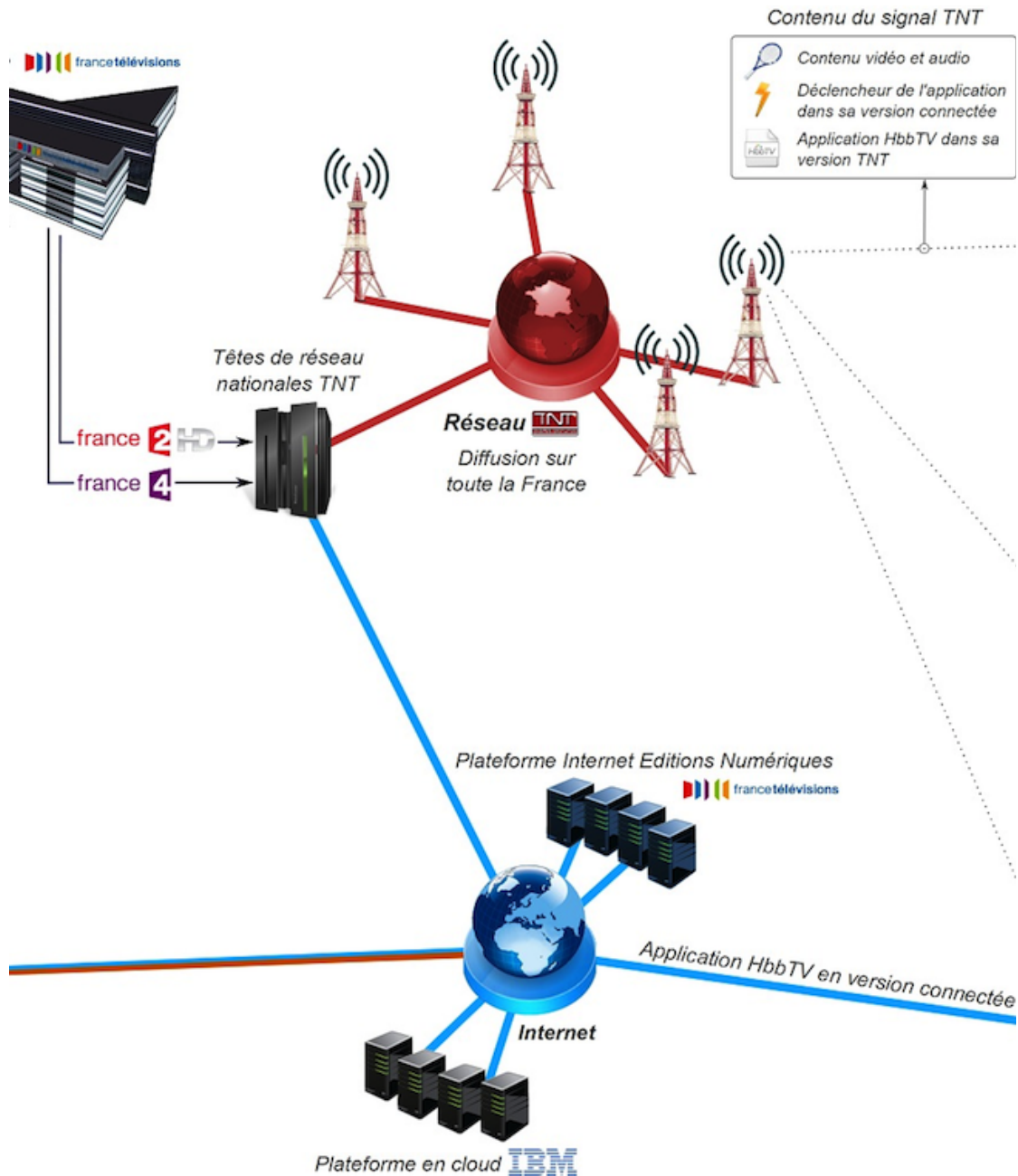


Figura 5.17 – HbbTV para Roland Garros. Detalle de el apartado de Difusión de las señales.

- Recepción: En la parte de recepción lo primero que se especifica es cual es el contenido de la señal que recibe el terminal. Este contenido es:
 - La propia señal de audio/video.
 - El disparo de la aplicación en televisores compatibles.
 - La aplicación HbbTV para televisores compatibles conectados.

Y luego distingue entre tres posibles escenarios en recepción:

- 1 – Televisión no compatible: El espectador no nota ninguna diferencia.
- 2 – Televisión compatible no conectada: El espectador recibe lo que ellos llaman la “versión TNT de la aplicación HbbTV”. Es decir, la aplicación por carrusel DSM-CC. TNT es el equivalente francés a TDT.
- 3 – Televisión compatible y conectada a Internet: Recibe la aplicación completa HbbTV que incluye un servicio de video bajo demanda con contenidos en 3D.

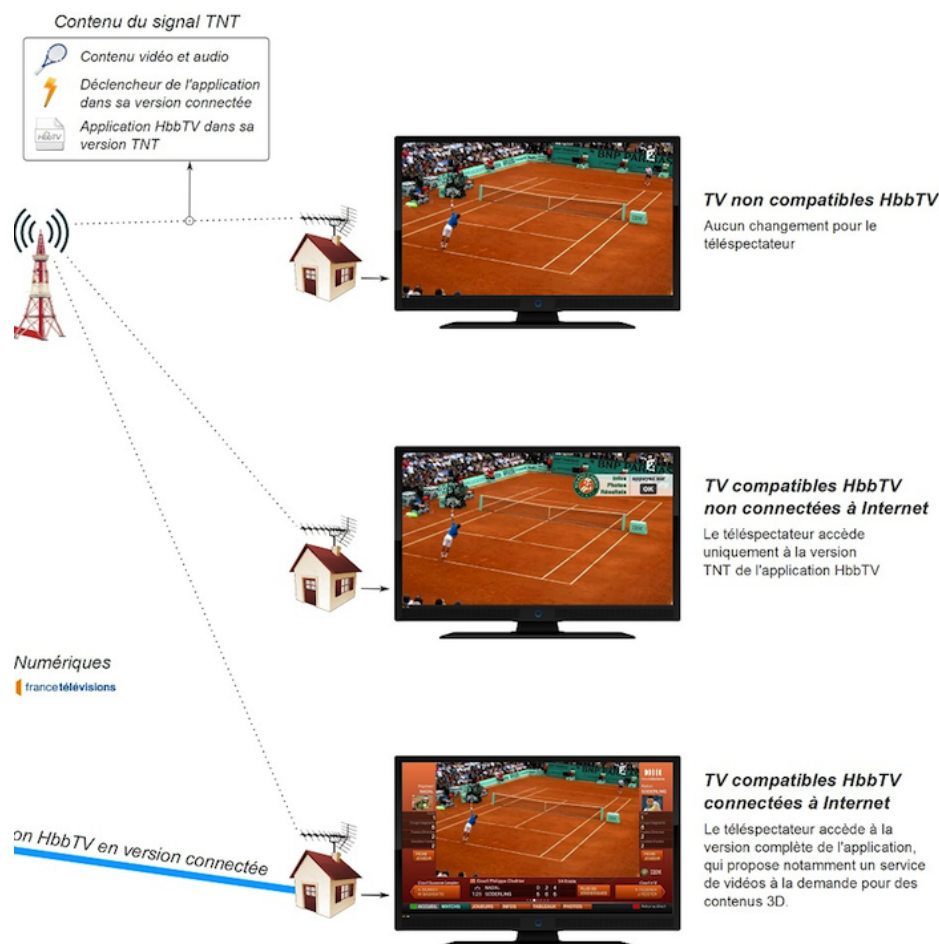


Figura 5.18 – HbbTV para Roland Garros. Detalle de el apartado de Recepción de la señal.

France Télévisions es el grupo que desde 2009 se encarga de la difusión de la televisión pública francesa. Es el mayor grupo de radiodifusión en Francia. Como se puede observar es un despliegue impresionante. Nos llevan unos 2 años de ventaja. ¿Conseguiremos superarles?...



6. GLOSARIO

6.1 Definiciones

Canal de retorno: Canal que permite el envío de datos desde el terminal hacia el broadcaster / proveedor de contenidos.

Middleware: Es una capa de software que de cara al resto del sistema se comporta como un driver. Se puede decir que es una capa de software que se comporta como un intermediario entre la aplicación y el resto del sistema. El resto del sistema puede ser la red, otras aplicaciones, el hardware.

Video bajo demanda: Video que un servidor proporciona a un cliente tras una petición de éste. El video se suele ofrecer en formato streaming.

Scripting: Lenguaje de programación que permite escribir programas que se ejecutan sin necesidad de ser compilados. Entornos de ejecución de un script pueden ser el sistema operativo (MS-DOS, Linux) o un programa (Navegador de internet).

Carrusel de objetos DSM-CC: Una agrupación lógica de datos que se transfieren usando el protocolo DSM-CC. Los datos se transmiten cíclicamente pero el terminal que los recibe los ve como si de un sistema de archivos se tratara gracias al carrusel de objetos y al protocolo DSM-CC.

Eventos (programación): Es una acción que se inicia fuera de los límites de un programa y que se gestiona con un trozo de código del interior del programa. Eventos pueden ser acciones de usuario (click, enter) o mensajes de otros programas.

Stream events: Un tipo de evento que viaja en el TS. La aplicación HbbTV debe estar preparada para detectarlo y procesarlo.

Servicio DVB: Es un conjunto de flujos multiplexados y relacionados entre si. Un servicio puede ser un canal (o más de uno) de televisión con un par de canales de audio, por ejemplo.

TV Anytime: Asociación de empresas que desarrollaba especificaciones para el almacenamiento digital de servicios audiovisuales. Es un proyecto que fue abandonado por 2005. La EBU últimamente ha rescatado los desarrollos que se hicieron en el apartado de metadatos.

Dispositivo híbrido: Dispositivo que cuenta con dos canales para la recepción de datos, el canal broadcast y el canal broadband.

D-Book: Libro de especificaciones técnicas para la TDT en el Reino Unido. Es publicado y mantenido por el DTG.

Set top box: Un terminal que se conecta a una televisión y que generalmente le proporciona una serie de servicios que la televisión no incluye originalmente. Un AppleTV es un STB y un sintonizador externo de TDT con HbbTV también lo es.

Catch-Up TV: Es un servicio que permite ver la programación ya emitida de un canal de televisión. Generalmente se suele proporcionar los últimos siete días de programación para ver por streaming. Este es otro ejemplo de VoD.



TV lineal: Es el servicio de televisión convencional. El canal de televisión que emite sus contenidos según una escaleta programada.

Framework: Un conjunto de librerías o clases que sirven para desarrollar aplicaciones. También puede ser un conjunto de métodos que permiten realizar una serie de funciones.

WebTV: Se refiere a la distribución de contenidos multimedia usando internet. Ejemplos de WebTV son Youtube, Hulu, Netflix.

Managed network: Es una red gestionada de operador que ofrece calidad de servicio. Por ejemplo la red de Ono.

Time-to-market: Se refiere al tiempo que se tarda en poner un producto o servicio en el mercado teniendo en cuenta la investigación, desarrollo, implementación y pruebas.

Servicios lineales: Un servicio lineal de televisión, se refiere a la programación convencional de un canal de televisión. Es equivalente a TV lineal o contenido A/V lineal.

Overlay: Sobreimpresión controlada o no, de unos contenidos sobre otros. No se distingue la procedencia del contenidos sobreimpresionado. Puede ser de internet o una aplicación local de la televisión.

Timeshift: Capacidad de un terminal de estar grabando la emisión de TV en una memoria temporal, dando al usuario la opción de poder pausar el programa o retomarlo desde un punto anterior.

TV Convencional: Lo mismo que servicios lineales o contenido A/V lineal. Se refiere a la televisión clásica en la que un programa comienza y termina según un horario predeterminado y sobre el cual el usuario no tiene ninguna capacidad de interacción.

Contenido A/V lineal: Ver TV convencional.

Streaming: La distribución de contenidos multimedia generalmente por redes TCP/IP en la cual el usuario consume el contenido al mismo tiempo que lo está viendo. Se diferencia de la descarga de contenidos en que no es necesario disponer previamente del contenido para poder verlo.

Múltiplex DVB: Es un conjunto de TS multiplexados formando uno o varios servicios de radiodifusión. Por ejemplo, el múltiplex de 7RM se encuentra en el canal 29 y transporta los servicios de 7RM, 7RM HD y Onda Regional.

Market / Google Play: Aplicación de los sistemas operativos Android de Google desde la cual se pueden descargar programas y contenidos para el sistema operativo. Funciona tanto en dispositivos móviles como en dispositivos Google TV.

iPhone: Teléfono móvil de la empresa Apple.

iPad: Tableta de la empresa Apple.

iOS: Sistema operativos de las tabletas y teléfonos móviles Apple.

iCloud: Sistema de almacenamiento de archivos en la nube de la empresa Apple.



Freeview/Freeview HD: Es la marca comercial de el conjunto de canales de TDT en el reino unido. Se fundó en el año 2002 y los accionistas son BBC, ITV, Channel 4, Sky y el operador Arquiva. Ofrece servicios interactivos (con MHEG-5) y es necesario un adquirir un terminal para poder disfrutarlo. Freeview+ añade capacidad de PVR y Freeview HD añade canales en alta definición.

JavaScript: Es un lenguaje de scripting interpretado más popular de internet. Funciona en los navegadores más populares Internet Explorer, Firefox, Chrome, Opera y Safari.

ECMAScript: Es un lenguaje de scripting estandarizado por ECMA en la especificación ECMA-262. Está basado en el JavaScript original, que ahora se considera un derivado de ECMAScript.

CEA-2014: Es un estándar diseñado por la CEA que ofrece un interfaz web y un framework para el control remoto de interfaces en dispositivos de electrónica de consumo. HbbTV utiliza indirectamente algunas partes de CEA-2014.

Web4CE: Es otra forma de denominar al estándar CEA-2014.

Parser: En lenguajes de programación es un analizador sintáctico que forma parte del compilador. Comprueba la sintaxis del código fuente y proporciona estructuras de datos al compilador.

Trickmodes: Se denomina así a las funciones disponibles para alterar el flujo de reproducción lineal de un contenido A/V. Estas funciones suelen ser pausa, avance rápido, retroceso rápido, parar.

h.264: Codec de vídeo de alta compresión ampliamente utilizado en entornos domésticos y profesionales, especialmente para hacer streaming de contenidos.

Bitrate: Tasa de bits. Es la cantidad de bits procesador por unidad de tiempo.

Hint track: Es una pista adicional de datos que se añade a los archivos para que ayuda al receptor en la reproducción.

Box / Atom: Son cada una de las unidades lógicas en las que se divide un archivo mp4 (box) o mov (atom).

Python: Lenguaje de programación interpretado.

Opencaster: Distribución Linux mantenida por la empresa Avalpa que incluye herramientas para codificar, encapsular, multiplexar y emitir flujos de A/V en TS multiplexados. También permite la generación de flujos de datos (DVB-SI, AIT, etc...)

Trikplay: Ver trickmodes.

Pseudo streaming: Tipo de streaming en el cual, en vez de consumir el contenido en el momento de su reproducción se hace una descarga por intervalos de una cantidad suficiente de datos como para que la reproducción sea continua..

Payload: Es el contenido que se transporta en una transmisión de datos. No incluye el "overhead" necesario para formar el resto del paquete y que llegue a su destino.



User-agent: Un user-agent es el programa que usa un cliente para el acceso a un recurso de red, servidor, etc... El user-agent puede ser un navegador, un cliente de correo, p.ej. A lo que solemos referirnos con user-agent es a la cadena de texto que ese user-agent utiliza para identificarse ante un recurso de la red. Por tanto deberíamos referirnos a él como "identificador del user-agent", pero se suele llamar user-agent directamente.

DVB-HTML: Es un estándar que permite el desarrollo de contenidos tipo web en televisiones digitales. Está basado en XHTML, CSS, DOM y ECMAScript.

6.2 Terminología

7RM	7 Región de Murcia
AAC	Advanced Audio Coding
AC3	Dolby Digital, Audio Coding 3
AEDETI	Asociación Española de Empresas de Televisión Interactiva
AIT	Application Information Table
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ARD	Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland – "Consortio de instituciones públicas de radiodifusión de la República Federal de Alemania
ASN.1	Abstract Syntax Notation One
ATSC	United States Advanced Television Systems Committee
AVC	Advanced Video Coding
B2B	Business-to-business
CA	Certification Authority
CDN	Content Delivery Network
CE-HTML	Consumer Electronics HTML
CEA	Consumer Electronics Association
CENC	Common Encryption
CSS	Cascading Style Sheets
CSS TV	CSS Perfil TV
DAE	Declarative Application Environment
DASH	Dynamic Adaptive Streaming over HTTP
DAVIC	Digital Audio Video Council
DOM	Document Object Model
DRM	Digital Rights Management
DSM-CC	Digital Storage Media - Command and Control
DTG	Digital TV Group
DTD	Document Type Definition
DVB	Digital Video Broadcasting
DVB-C	DVB-Cable
DVB-S	DVB-Satellite
DVB-T	DVB-Terrestrial
EBU	European Broadcasting Union
EIT	Event Information Table
EIT p/f	EIT present/following
EPG	Electronic Program Guide
ETSI	European Telecommunications Standards Institute
FNMT	Fábrica Nacional de Moneda y Timbre



FQDN	Fully Qualified Domain Name
GEM	Globally Executable MHP
GET	Método HTTP para recibir datos
GOP	Group of Pictures
HbbTV	Hybrid Broadcast Broadband Television
HD	High Definition
HE-AAC	High-Efficiency AAC
HTML	Hypertext Markup Language
HTML4	HTML v4
HTML5	HTML v5
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IFA	Internationale Funkausstellung (Feria Internacional de servicios de Broadcast)
IP	Internet Protocol
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
ISOBMFF	ISO Base Media File Format
M2TS MPEG-2	Transport Stream
MHEG	Multimedia and Hypermedia information coding Expert Group
MHP	Multimedia Home Platform
MOOV	movie atom/box
MP2	MPEG-1 Audio Layer 2
MP3	MPEG-1 Audio Layer 3
MP4	MPEG-4 Part 14
MPD	Media Presentation Description
MPEG	Motion Picture Experts Group
NAT	Network Address Translation
OIPF	Open IPTV Forum
OTT	Over the top
PAT	Program Association Table
PES	Packetized Elementary Stream
PID	Packet Identifier
PMT	Program Map Table
POST	Método HTTP para enviar datos
PS	Program Stream
PSI	Program-specific information
PVR	Personal Video Recorder
RFC	Request For Comments
RTCP	Real Time Control Protocol
RTRM	Radiotelevisión de la Región de Murcia
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RTVE	Radiotelevisión Española
SD	Standard Definition
SDK	Software Development Kit
SDP	Session Description Protocol
SDT	Service Description Table
SETSI	Secretaría de Estado para las Telecomunicaciones y la Sociedad de la Información
SGML	Standard Generalized Markup Language



TCP	Transmission Control Protocol
TLS	Transport Layer Security
TS	Transport Stream
UDP	User Datagram Protocol
UPNP	Universal Plug and Play
URL	Uniform Resource Locator
VoD	Video on Demand
W3C	World Wide Web Consortium
XHTML	Extensible HyperText Markup Language
XML	eXtensible Markup Language



7. Bibliografía

BLOQUE 1

AEDETI – Requisitos de servicios interactivos v2.0.7
BBC - MHP White Paper
BBC – ITV Design
www.usabilityblog.de
Deutsche TV Platform - Hybrid TV White book
DigiTAG - DiscussionPaper v1.0 - The implementation of hybrid TV services in Europe
EBU - Principles for connected and hybrid TV
EBU – Viewpoint The future is Hybrid
Estándares
 ETSI ES 202 184 c2.1.1 MHEG-5
 Youview Core Technical Specification
Documentación Grupo de estudio de la televisión híbrida
HD Forum Symposium Diciembre 2011
 Introduction to HbbTV – Klaus Illgner-Fehns (IRT)
 HbbTV in Spain: a brief update – José Luis Vázquez (AEDETI)
 Merging linear with on demand – Bram Tullemans (EBU)
 Opening the way to HbbTV in Spain – Francisco Asensi (RTVE)
Hbbtv – Press Release HbbTV 1.5
IFA – Trends 2012
IRT – HbbTV Workshops
IRT – Hybride TV Empfänger Jun09 – Klaus Merkel (IRT)
IRT – Status und Ausblick Dic10 – Klaus Merkel (IRT)
IRT – Status und Ausblick Dic11 – Klaus Merkel (IRT)
Impala white paper MHEG-5
EBU - Technical Review HbbTV 2010Q1 – Klaus Merkel (IRT)
Estado actual del estándar MHP Feb2005 – TVC Multimedia
Mediatvcom - HbbTV Explained
Connected TV Developments May2011 – Colin Prior (S&T)
SETSI – Receptores TDT interactivos Propuesta definitiva
Connected TV (CTV) Standardisation in the UK – Steve Morris (DTG HTML Group)
W3C Workshop - Web and Tv Feb2012
WBU - Hybrid and Internet Television
White Paper An interactive television advertising overview Dic2011 – IAB



BLOQUE 2

ISO 23009-1 – DASH
ISO 23001-7 – Encriptación ISOBMFF
ANSI CEA 2014 A
Avalpa - Opencaster User Manual v3.0
DAVIC 1.4.1 Specification part 9
IRT Adaptive Streaming Tests 2011
W3C – CSS Color Module Level 3
W3C – CSS Mobile Profile 2
W3C – CSS TV Profile 1
W3C – DOM2-Core
W3C – DOM2-Events
W3C – DOM2-HTML
W3C – DOM2-Style
DVB Work on Multimedia Transport
Developers Garden Techtalk HbbTV – Stefan Merten (Capgemini)
EBU - Tech3350 Subtitling Format Definition
EBU - HBB workshop 09 kozamernik
EBU HBB EBU Requirements – Franc Kozamernik (EBU)
ECMA-262, 3rd edition, December 1999
ECMA-262 5.1 edition, June 2011
HBBNEXT – Public Consultation FP7 ICT Work Programme Ene12 – Ralf Neudel
HbbTV Errata
HbbTV Specification 1.5
Http - ProductSheet httpStream
Content sharing and experience sharing with Web4CE – Lowet, Shrubsole (Philips)
IRT – HbbTV Test Harness
ISMA 2.0
ISO IEC 23001-6 Dynamic adaptive streaming over HTTP (DASH)
ISO IEC 23009-1 DASH Part1
OIPF Release 1 – Volume 2 – Media Formats
OIPF Release 1 – Volume 4 – Protocols
OIPF Release 1 – Volume 5 – DAE
Recording-and-Delivery-of-HbbTV-Applications – Dufourd, Thomas, Concolato (Telecom Paris Tech)
W3C-Workshop – DASH – Stockhammer (Qualcomm)
ITU – IPTV Multimedia Application Platforms – Samsung Proposal
Tektronix MPEG Fundamentals



DASH – Timmerer (AAU), Müller (ITEC)
W3C – OIPF DAE – Lindquist (Ericsson)
W3C – XHTML 1.0 Reformulation of HTML4 in XML 1.0
W3C – XMLHttpRequest Level 2
Web4CE: Accessing Web-based Applications on Consumer Devices – Dees, Shrubsole (Philips)
ATSC Standard - Digital Audio Compression (AC-3, E-AC-3)
EBU – MHP in Italy – Alberico (RAI)
QuickTime File Format Specification – Apple Developer
RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels
RFC 2116 – HTTP 1.1
RFC 2246 – TLS v1.0
RFC 2326 – RTSP
RFC 2327 – SDP
RFC 2818 – HTTP over TLS
RFC 3550 – RTP
RFC 3551 – RTP Profile for Audio and Video Conferences with Minimal Control
RFC 3640 – RTP Payload Format for transport of MP4 Elementary Streams
RFC 3984 – RTP Payload Format for H.264 Video
RFC 4346 – TLS v1.1
RFC 5246 – TLS v1.2
RFC 5280 – PKI x.509
RFC 5746 – TLS Renegotiation Indication Extension
ETSI TR 101 202 v1.2.1 – Implementation guidelines for Data Broadcasting
ETSI TS 102 034 v1.4.1 – Transport of MPEG-2 TS Based DVB Services over IP Based Networks
ETSI TS 102 366 v1.2.1 – Digital Audio Compression (AC-3, Enhanced AC-3) Standard
ETSI TS 102 539 v1.3.1 – Carriage of Broadband Content Guide (BCG) information over IP
ETSI TS 102 809 v1.1.1 – Signalling and carriage of interactive applications and services in Hybrid broadcast/broadband environments
ETSI TS 101 154 v1.9.1 Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream
ETSI TS 102 796 v1.1.1 – HbbTV
ETSI EN 300 468 v1.5.1 – Specification for Service Information (SI) in DVB systems



BLOQUE 3

EBU – Tech-i Issue 4 June 2010

Ayuda de ffmpeg

Manuales Ubuntu Server

Manual Flash Media Server

Manual Wowza Streaming Server

OIPF Release 1 DAE Reference Guide

OIPF Release 1 – Volume 5 - DAE

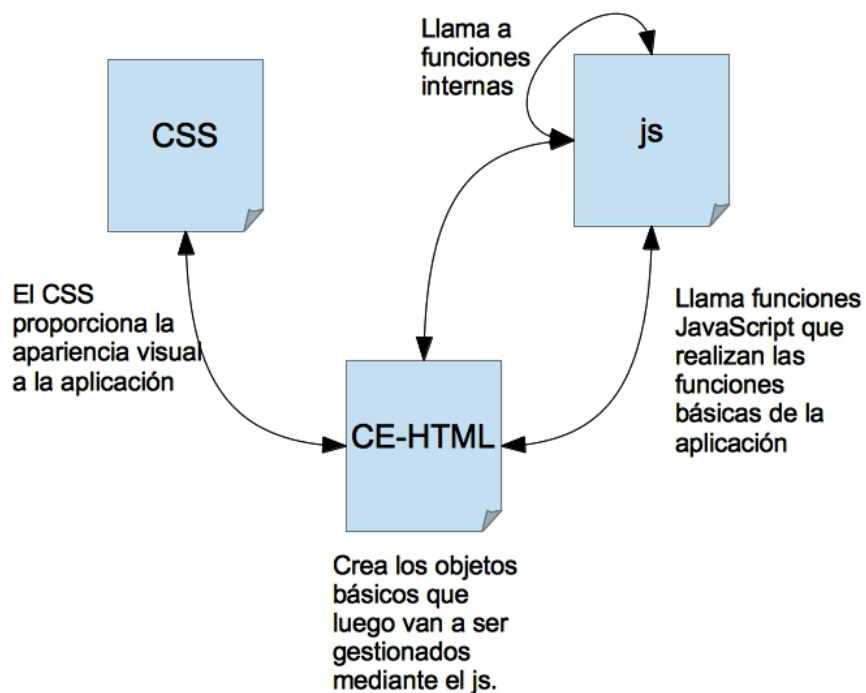


ANEXO A – Estructura básica de una aplicación HbbTV

En este anexo vamos a exponer brevemente como es la arquitectura de una aplicación básica y vamos a resaltar los trozos importantes en su estructura.

Las aplicaciones pueden seguir distintas arquitecturas. La más utilizada suele ser un archivo CE-HTML que tiene las funciones en un archivo JavaScript independiente y que controla su apariencia visual con un archivo CSS también independiente. El archivo CE-HTML importa estos archivos (normalmente al inicio) y ya puede usar las funciones y aplicar estilos con cada objeto que se crea en el código CE-HTML.

El esquema sería este:



Vamos a mostrar cuales son los componentes necesarios para el archivo CE-HTML y su contraparte en las funciones del archivo JavaScript.



CE-HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//HbbTV//1.1.1//EN" "http://www.hbbtv.org/dtd/HbbTV-1.1.1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>HbbTV</title>
  <meta http-equiv="cache-control" content="no-cache">
  <meta http-equiv="expires" content="0">
  <meta http-equiv="pragma" content="no-cache">
  <meta http-equiv="Content-Type" content="application/vnd.hbbtv.xml+xhtml; utf-8" />
  <script type="text/javascript" src="hbbtv.js"></script>
  <link rel="stylesheet" href="hbbtv.css" />
</head>

<body onload="init();">
  <div style="visibility: hidden; display: none;">
    <object type="application/oipfApplicationManager" id="applicationManager" style="position:absolute;
left:0px; top:0px; width:0px; height:0px;">
    <object type="application/oipfConfiguration" id="configuration" style="position:absolute; left:0px;
top:0px; width:0px; height:0px;">
  </div>
  <div>Area disponible para la aplicacion HbbTV: 1280x720
    <div id="area_segura"> Area segura: 1152x648
      <div id="info"></div>
      <div class="text"></div>
      <div id="pagina_inicial">
        <a id="boton_rojo" class="button">Boton Rojo</a>
      </div>
      <div id="aplicaciones">
        <a id="boton_verde" class="button">WWW.7RM.ES</a>
        <a id="boton_amarillo" class="button">CITA MÉDICA</a>
        <a id="boton_azul" class="button">Reproductor de video</a>
      </div>
    </div>
  </div>
</body>
</html>
```

Esta es la forma de identificar correctamente una aplicación XML HbbTV.

Esta cabecera HTTP también es necesaria para identificar el contenido HbbTV.

La creación de estos dos objetos es imprescindible para el correcto funcionamiento de la aplicación. Se crean dos objetos específicos de hbbTV:

- 1 – El Application Manager
- 2 – Configuration

El primero ya hemos comentado que sirve para controlar el ciclo de vida de la aplicación.

El segundo sirve para reportar las configuraciones del terminal a la aplicación.

El objeto se crea en el CE-HTML pero se gestiona con funciones JavaScript.

Estos objetos son específicos de HbbTV. Un navegador normal no sabe interpretarlos.



JAVASCRIPT

La siguiente función forma parte del fichero JavaScript hbbtv.js. Ese archivo tiene muchas más funciones y es bastante amplio, pues es la “inteligencia” de la aplicación. De entre todas las funciones, la aplicación hbbtv.html (CE-HTML), llama a esta función nada más cargarse. Al lado de cada línea hay un comentario explicando su función.

```
function init() {
    document.getElementById("boton_rojo").focus(); //Pone el foco en el botón rojo
    try {
        var appManager = document.getElementById("applicationManager"); //Se asigna a la variable
        appManager el objeto ApplicationManager que hemos creado en el CE-HTML.
        currentApp = appManager.getOwnerApplication(document); //Enlazamos el documento CE-HTML a
        nivel DOM con el ApplicationManager. A partir de ahora podemos controlar el ciclo de vida de la aplicación y
        crear y destruir nuevas aplicaciones.
        currentApp.show(); //Mostramos la aplicación.
        remoteControlKeys = currentApp.privateData.keySet; //Crea la variable remoteControlKeys y le asigna
        todo el juego de teclas disponible por la aplicación y...
        remoteControlKeys.setValue(remoteControlKeys.RED); //...de entrada limitamos (aplicamos una
        máscara) para que el único botón que se pase a la aplicación sea el botón rojo, si no, al apretar una tecla
        que no esté en pantalla, podría hacer algo. Esto es algo derivado del diseño de nuestra aplicación no tiene
        por que ser necesario en cualquier aplicación, aunque si suele ser habitual. Pero hasta que no registremos
        un manejador de evento esto no va a funcionar.
    }
    catch (e) {
        document.getElementById('info').innerHTML = 'Error en init()'; //En caso de un error lo indicaríamos
        poniendo este texto en el objeto “info”.
    }
    apps = document.getElementById("aplicaciones"); //Asignamos a la variable apps una sección del CE-
    HTML. No es relevante.
    document.addEventListener("keydown", handleKeyCodes, false); //Este es el manejador de evento
    necesario para que la aplicación pase las pulsaciones de teclas a la función que gestiona las pulsaciones de
    las teclas. Concretamente recoge eventos de "keypress" y los pasa al listener, en este caso
    handleKeyCodes, que debe ser o un objeto que implemente el interfaz KeyListener o una función
    JavaScript. En este caso es una función handleKeyCodes(). El false indica que el manejador no activa el
    modo captura de eventos. Concretamente, se recoge un evento “keypress” y se pasa a una función
    handleKeyCodes.
}
```



Esto es parte de la función `handleKeyCodes()`; que realiza acciones según la tecla pulsada del mando a distancia. Esta aplicación ejecutada en un ordenador no funcionaría porque las teclas que se monitorizan son teclas específicas de un mando a distancia.

```
function handleKeyCodes(e) { //Se le pasa como evento la tecla pulsada
    switch (e.keyCode) {

        /* Una pulsación del botón rojo llama a otra función, toggleApps(); que lo que hace
        es mostrar o esconder los botones verde, amarillo y azul*/
        case KeyEvent.VK_RED: //En caso de ser una pulsación del botón rojo
            document.getElementById("boton_rojo").focus(); // situar el foco sobre el enlace
            activarApps(); // y llamar a la función toggleApps();
            break;

        /* Con el botón verde realizamos otra acción distinta. Se llama a
        createApplication(URL,boolean) y crea una nueva aplicación como hija (true) de la
        aplicación que la crea. Al ser una aplicación con otro FQDN estamos creando una
        aplicación broadcast-independent. Estamos abriendo la página web de 7RM.*/
        case KeyEvent.VK_GREEN:
            document.getElementById("boton_verde").focus();
            break;

        /* Al igual que arriba, crea una aplicación pero esta vez la aplicación se crea como
        hermana (false) de la aplicación creadora. Estamos abriendo la página de la cita
        médica del Servicio Murciano de Salud*/
        case KeyEvent.VK_YELLOW:
            document.getElementById("boton_amarillo").focus();
            currentApp.createApplication('https://sms.carm.es/cmap/iniciarReserva.do',false);
            break;
    }
}
```



```
/* Con la pulsación del botón azul se abre otra página con mismo FQDN de la
página original (la página sigue siendo broadcast-related, si es que lo era
originalmente). Concretamente abre una página con un reproductor de vídeo en la
que podemos hacer play a un video descargado por streaming HTTP.*/
case KeyEvent.VK_BLUE:
document.getElementById("boton_azul").focus();
window.location.replace("Reproductor.html");
break;
/* Cuando estemos en la ventana del reproductor de vídeo
la tecla PLAY del mando a distancia estará habilitada y el video se reproducirá. Se
hace una llamada a una función (playVideo();) que en el interior de su código realiza
la reproducción del vídeo. Esto no es la función que reproduce vídeo
directamente. La función playVideo() también está en el archivo JavaScript.*/
case KeyEvent.VK_PLAY:
playVideo();
break;
// Idem PLAY
case KeyEvent.VK_PAUSE:
pauseVideo();
break;
// Idem PLAY
case KeyEvent.VK_STOP:
stopVideo();
break;
/*Si pulsamos el botón "OK" de mando a distancia finalizaremos la aplicación con
la función currentApp.destroyApplication();*/
case KeyEvent.VK_ENTER:
currentApp.destroyApplication();
```

Hay mas código dentro de esta función pero estas son las funciones relevantes en las que se puede observar como gestionar eventos de teclas del mando a distancia. Recordamos que para que esta función evalúe los eventos de teclado del mando a distancia primero hay que registrar el manejador de eventos de teclado y luego hay que permitir que se pase a la aplicación las teclas que nos interesan. Esto lo hemos hecho en la función init().



Repasemos que conseguimos con los extractos de código anteriores:

1 – Declarar los objetos `ApplicationManager` (objeto heredado del DAE) y `Configuration` (objeto heredado del DAE) en el cuerpo de la página CE-HTML. Estos son necesarios para controlar el ciclo de vida de la aplicación y para poder responder con las capacidades del terminal si el servidor de la aplicación necesita saberlas.

2 – Inicializamos la conexión entre el `ApplicationManager` y un objeto JavaScript que nos permita controlar el ciclo de vida de la aplicación. Es decir, cuando se termina la aplicación, cuando se inicia, cuando se muestra y cuando se esconde.

3 – Hemos visto como iniciar un manejador de eventos de teclado para que las pulsaciones del mando a distancia se pasen a la aplicación.

4 – También hemos visto como se ha leído todo el conjunto de teclas disponibles y se ha filtrado solo el uso a la que nos interesa.

5 – Hemos visto como se manejan las teclas en una función auxiliar (`handleKeyCodes(e)`) que recibe como eventos de teclado lo que le envía la función que hemos registrado para manejar los eventos de teclado.

6 – Dentro de la función que maneja los eventos de teclado hemos visto varias opciones de creación y destrucción de aplicaciones.

A – Hemos visto como crear una aplicación hija con `application.createApplication(URL, boolean)`. Se hace con el `boolean=true`.

B – Hemos visto como crear una aplicación al nivel de la aplicación creada (sibling según el DOM). Se hace con el `boolean = false`.

C – Como abrir otra página dentro del mismo dominio con `window.location.replace(página)`. Con esto iremos pasando por las distintas páginas que componen nuestra aplicación.

D – Como finalizar una aplicación con `application.destroyApplication()`.

A continuación vamos a ver pequeños trozos de código para añadir funcionalidad a las aplicaciones.



Anexo B – Funciones básicas para aplicaciones HbbTV

Reproducir un vídeo por streaming HTTP

Declarando este objeto en el código CE-HTML:

```
<object id="video" data="http://192.168.1.5/HbbTV/Demo.mp4" type="video/mp4" style="width:720px; height:405px;"></object>
```

estamos creando un objeto de video tipo MP4, con una URL y un tamaño predeterminado. Al objeto lo identificamos de cara al DOM como “video”. Es decir, luego desde el JavaScript llegaremos al objeto mediante este identificador y podremos trabajar con él.

La URL es la de un servidor HTTP que tiene instalados los plugin para hacer pseudostreaming.

El Javascript que maneja este objeto de video con identificador “video” es este:

1º – Al igual que con el ApplicationManager mapeamos (unimos) el objeto video del CE-HTML con un objeto del DOM que podamos manejar con Javascript. Esto se hace asignando el objeto llamado “video” en el CE-HTML a una variable también llamada video en el código Javascript:

```
var video = document.getElementById("video");
```

Esto se puede hacer en una función init() para la página del reproductor de video, p.ej.

Teniendo ya el objeto video CE-HTML metido en la variable video DOM.

1 – Para reproducir → *video.play(1);*

2 – Para pausar → *video.play(0);*

3 – Para parar la reproducción → *video.stop();*

Estos y otros métodos para la reproducción de contenido streaming están listados en la Tabla 4.3.



Comprobar el estado de reproducción de un video

Con el objeto de video asignado a la variable “video”, asignamos el estado de reproducción del video a una nueva variable “estado_video” de la siguiente forma:

```
var estado_video = video.playState;
```

y, en función del valor devuelto:

case 0: // **Parado**

```
document.getElementById('stop').focus();  
estado = 'Estado del video (0): Parado';  
break;
```

case 1: // **Reproduciendo**

```
document.getElementById('play').focus();  
estado = 'Estado del video (1): Reproduciendo';  
break;
```

case 2: // **Pausado**

```
document.getElementById('pausa').focus();  
estado = 'Estado del video (2): Pausado';  
break;
```

case 3: // **Conectando**

```
estado = 'Estado del video (3): Conectando con el servidor';
```

case 4: // **Almacenando**

```
estado = 'Estado del video (4): Almacenando antes de la reproduccion (buffer);  
break;
```

case 5: // **Terminado**

```
document.getElementById('volver').focus();  
estado = 'Estado del video (5): Se ha terminado la reproduccion del video';  
break;
```



El último caso, el 6, se corresponde con un error en la reproducción. En función del código de error:

```
case 6: // Error

    stopVideo(); //Paramos la reproducción del video
    estado = 'Estado del video (6): Ha habido un error durante la reproduccion';

    switch (video.error) {

        case 0:
            error = Error (0) - Formato A/V no soportado';
            break;

        case 1:
            error = Error (1) - No se puede conectar con el servidor o conexion perdida';
            break;

        case 2:
            error = 'Error (2) - Error no identificado';
            break;
    }

    break;
```

Esta última función es muy útil en las fases de desarrollo por que si por algún motivo no se reproduce el video puede ayudarnos a identificar el motivo.

Incluir un video broadcast en la aplicación

Creamos el objeto en el código CE-HTML:

```
<object id="broadcast" type="video/broadcast" style="width:100%; height:100%;"></object>
```

Con esto tendríamos la señal de TV en la subdivisión que hayamos preparado para ello dentro de la aplicación. En este caso ocuparía el 100% del espacio asignado. Al igual que con el video broadband por streaming podemos asignar el objeto identificado como “broadcast” a una variable y cambiarlo de posición, detenerlo, etc. con métodos JavaScript heredados del DAE (punto 4.5.2.1 de este proyecto).



Obtener las capacidades del terminal

De la misma forma en la que hemos estado creando objetos ApplicationManager (oipfApplicationManager) y objetos Configuration (oipfConfiguration) para obtener las capacidades de un terminal debemos crear un objeto Capabilities (oipfCapabilities) en el código CE-HTML. Recordemos que el terminal internamente pasa un archivo formateado en XML para indicar sus capacidades (ver punto 4.9.3 de este proyecto).

```
<object id="objeto_capacidades" type="application/oipfCapabilities" style="position: absolute; left: 0px; top: 0px; width: 0px; height: 0px;"></object>
```

A partir de este objeto podremos obtener las capacidades del terminal con funciones Javascript.

```
var capacidades = document.getElementById('objeto_capacidades'); //Asignamos el objeto oipfCapabilities a la variable capacidades.
```

- Mostrar las capacidades del terminal:

```
xmllp = capacidades.xmlCapabilities; //Asignamos a la variable xmllp la lista de capacidades obtenida  
var serializer = new XMLSerializer(); // Creamos un objeto para poder convertir a una cadena de texto el XML recibido  
xmlltxt = serializer.serializeToString(xmllp); //lo convertimos a texto y lo guardamos en la variable xmlltxt
```

- Preguntar por las capacidades adicionales, teniendo el objeto oipfCapabilities mapeado a la variable "capacidades":

RTSP: `capacidades.hasCapability(+RTSP)` devolverá `true` si está presente y `false` si no lo está.

Descarga: `capacidades.hasCapability(+DL)` devolverá `true` si está presente y `false` si no lo está.

Grabación: `capacidades.hasCapability(+PVR)` devolverá `true` si está presente y `false` si no lo está.

Obtener la configuración del terminal

Con esta función podemos saber como tiene el usuario configurado el terminal. Si por ejemplo, lo tiene para audio en inglés, o si tiene subtítulos, o en que país está. Esto lo podemos usar para personalizar el contenido que le vamos a entregar a sus preferencias locales, p.ej, la aplicación en otro idioma.

Creamos en el código CE-HTML el objeto que nos permitirá acceder a esta información:

```
<object id="objeto_configuracion" type="application/oipfConfiguration" style="position: absolute; left: 0px; top: 0px; width: 0px; height: 0px;"></object>
```

Mapeamos el objeto a una variable que podamos utilizar con Javascript:

```
var configuracionoipf = document.getElementById('objeto_configuracion');  
var configuracion = oipfcfg.configuration; //esta es la variable que vamos a usar para obtener la información.
```

- Mostramos idioma de audio preferido: `resultado = configuracion.preferredAudioLanguage;`
- Mostramos idioma preferido para los subtítulos:
`resultado = configuracion.preferredSubtitleLanguage;`
- Mostramos el país configurado en el terminal: `resultado = config.countryId;`



Mostrar la información del EPG

Esta función nos permitirá mostrar la información de EPG (Electronic Program Guide) que tiene el canal broadcast. Por lo tanto trabajamos con un objeto video/broadcast. En este caso no necesitamos nada del OIPF directamente.

Partiendo del objeto de video broadcast:

```
<object id="broadcast" type="video/broadcast" style="width:100%; height:100%;">
```

Creamos la siguiente función:

```
function Eitpf(indice) { // indice es un array
  var video = document.getElementById('broadcast'); // mapeamos el objeto broadcast a la variable video
  var evento = video.programmes[indice]; // consultamos los programas por su índice (0 actual, 1 próximo)
  var dia_comienzo = new Date(evento.startTime * 1000); // sacamos el día de comienzo con su hora
  var dia_fin = new Date((evento.startTime+evento.duration) * 1000); // sacamos el día de fin
  var desde_horas = dia_comienzo.getHours(); // del día de comienzo sacamos la hora de inicio
  var desde_minutos = dia_comienzo.getMinutes(); // los minutos
  var hasta_horas = dia_fin.getHours(); // y del día de finalización sacamos la hora de finalización
  var a_minutos = dia_fin.getMinutes(); // y los minutos
  return (desde_horas<10?'0:')+(hasta_horas+(desde_minutos<10?'0:')+(desde_minutos+' - '+
(hasta_horas<10?'0:')+(hasta_horas+(a_minutos<10?'0:')+(a_minutos+'<br />'+evento.name;
} // devolvemos una cadena formateada
```

Con esta función obtenemos todos los datos de tiempo y nombre de programa del programa actual (índice 0) y del programa siguiente (índice 1) y devolvemos una cadena formateada para que la siguiente función pueda mostrar la información:

```
var txt = 'Ahora: '+getEitEvent(0)+'<br />Seguido por: '+getEitEvent(1);
```

La salida de esta función es así:

Ahora: 21:30 – 22:00

MUSICAL 7

Seguido por: 22:00 – 00:00

CINE 7RM