



# Curso Multimedia Home Platform 1.1.2

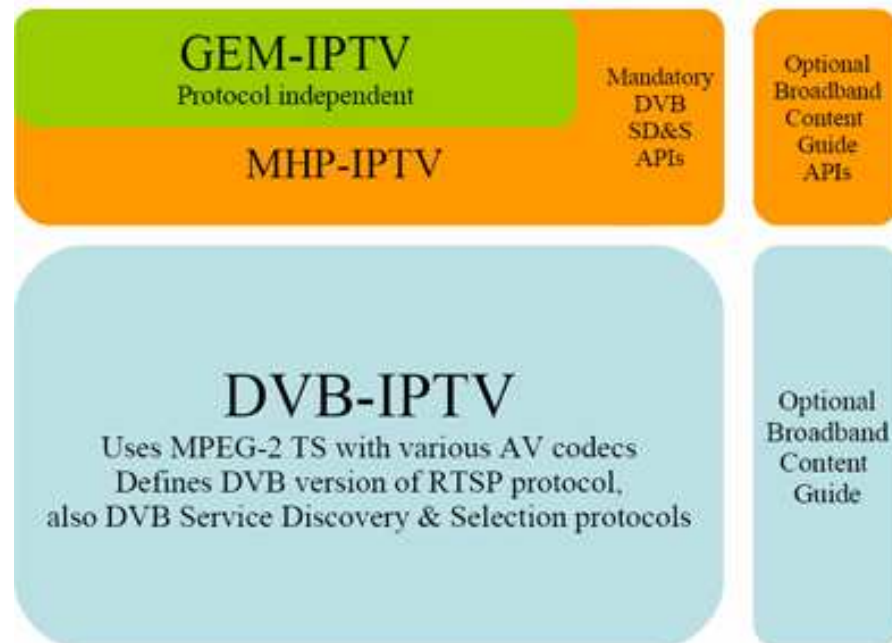
MHP 1.1.3 y LO QUE VIENE

## Introducción

- Si visitamos la web de mhp: [www.mhp.org](http://www.mhp.org), observaremos que nos espera un futuro prometedor respecto a MHP.
- Para empezar vemos que las especificaciones MHP van a la par que las de GEM, respecto a lo cual ya hablamos en el primer capítulo, siendo los tres grupos existentes los siguientes:
  - MHP/GEM 1.0
  - MHP/GEM 1.1
  - MHP/GEM 1.2
- Hoy hay implementaciones del segundo grupo, disponiendo de implementaciones de MHP 1.1.2. Pero aún no hemos visto ninguna de 1.1.3.
- Igualmente de DVB-HTML no he visto ninguna implementación en un STB, aunque ortikon ([www.ortikon.com](http://www.ortikon.com)) dispone de un producto que renderiza DVB-HTML

## Introducción

- En el tercer grupo, MHP/GEM 1.2, nos encontramos con las especificaciones
  - MHP 1.2
  - GEM 1.2
- Que así no nos dicen nada, pero si explicamos que estos perfiles incluyen las configuraciones para IPTV la cosa cambia.



tam1032r1-mhp-iptv-presentation.pdf

## Introducción

- Antes de continuar. Lo que vamos a ver a continuación es un revisión/resumen del siguiente documento:

**`tam1032r1-mhp-iptv-presentation.pdf`**

El cual podéis encontrar en la web de mhp.org en el link:

<http://www.mhp.org/docs/tam1032r1-mhp-iptv-presentation.pdf>

## MHP 1.1.3

- Es la última versión del grupo MHP 1.1.x. Veamos los elementos nuevos y mejoras con respecto a la última especificación 1.1.2.
- **Smartcards**
  - Como ya descubrimos en el capítulo referido a SATSA, en la versión 1.1.3 se incluyen mejoras en este sentido.
  - Se añade una nueva gestión de eventos para saber cuando una tarjeta se inserta y se saca del dispositivo. Esto se ha incluido después de la experiencia en Italia.
  - **Se incluye un tipo de comunicación denominada “raw” para aquellos dispositivos que no soportan SATSA.** Entendemos que para dar cabida a hardware “Legacy”.
  - Se actualiza el modelo de gestión del “Cryptographic Service Provider” con otro más genérico. Ahora más de una aplicación puede acceder al servicio pero cada una obtiene su copia de las clases.

## MHP 1.1.3

- **Gráficos**

- La resolución **1920x1080** deja de ser obligatoria para ser opcional.
- Se añade una nueva resolución de **960x540** para píxeles cuadrados, con el fin de ofrecer compatibilidad con **OCAP**.
- En la AIT las aplicaciones podrán informar de las **configuraciones gráficas que soportan** y de sus comportamientos ante determinadas situaciones, en concreto:
  - Qué resoluciones soporta
  - Comportamiento de la APP si no soporta las resoluciones disponibles
  - Comportamiento de la APP cuando el Video del Service es escalado por la EPG o el Navigator.
- Se mejora la integración del PBP en MHP: respecto a Graphics2D y DVBGraphics, y también en cuanto a Inter-Xlet Communication: IxcPermission

## MHP 1.1.3

- **I-Frames**
  - No están obligadas a soportar I-Frames en modo HD.
- **Se “Limpian” diferentes APIS**

### **Service Selection API**

- `ServiceContext.select(Locator[])` and applications
- Re-selecting the currently selected service

**OJO: Java TV 1.0 se reemplaza por Java TV 1.1**

### **Stored applications**

- Se pueden guardar APPs de forma síncrona o asíncrona
- Se pueden almacenar APPS que no estén en el `AppsDatabase` en un `StoredApplicationService`
- Re-escritura de los requerimientos de seguridad

## MHP 1.1.3

- **Se “Limpiar” diferentes APIS**

  - **Component based players**

    - “Atomic video swap between background and component based players”: ¿?
    - Se puede hacer Service selection cuando el Video del Service se está presentando en un **Component Based Player**

- **Pasamos de Java TV 1.0 a Java TV 1.1 (JSR-927)**

  - La mayor parte de los errores de la 1.0 se resuelven en la 1.1

- **Otros**

  - Se solucionan pequeños problemas encontrados por **implementadores** de OCAP y Blu-ray.
  - El API de acceso a la **memory card** se actualiza hasta el de la última versión de OCAP
  - El requerimiento de tamaño de claves baja de 4096 a 2048

## MHP 1.2

- Veamos los elementos más relevantes de la versión 1.2
- **Unbound Applications**
  - En las versiones 1.0.x sólo se pueden transmitir APPs signaled en el Broadcast
  - En algunos casos, como el Navigator o la EPG, es necesario que las APPs sean horizontales.
  - Esto se “solucionaba” haciendo signalling de la app en todos los Services...
  - **OCAP** soluciona el problema con las “**Unbound Applications**”, las cuales se ejecutan de forma independiente al Broadcast. El STB obtiene la lista de Apps Unbound del Signalling de la network.
  - MHP adopta **parte** de lo definido por OCAP:
    - Se adopta la arquitectura
    - Descriptores de OCAP para Abstract Services y Unbound APPS
    - org.ocap.service.AbstractService

## MHP 1.2

- **Unbound Applications**

- Funcionalidad de OCAP incluida pero con diferente solución técnica:
  - Las Unbound APPS **vian en un DVB Service específico**.
  - El almacenamiento de las Unbound ha de ser explícito en lugar de basado en prioridades para caching. Se re-utiliza el paquete `org.dvb.application.storage`
  - Las prioridades en cuanto a recursos ahora son a nivel de Service Context en lugar de APP.
- NO se ha incluido el control de Recursos de Aplicaciones entre Abstract Services y Unbound Applications.

- **Providers, Adaptadores y Traductores de Protocolos**

- Un Framework de Providers permite a partes de los APIS Standard trabajar con protocolos que no son soportados en el middle del STB.
- Esto sucede por ejemplo cuando se definen protocolos después de la venta de los STBs, o si la interpretación de cómo debe funcionar un determinado protocolo standard puede ser muy divergente...

## MHP 1.2

- **Providers, Adaptadores y Traductores de Protocolos**

- Aquellos Providers que formen parte de las aplicaciones MHP se pueden proporcionar **sin necesidad de System Updates** (vamos, sin actualizar el software del STB) y se pueden almacenar en el deco utilizando el API de **Application Storage**.
- Existen **dos tipos de Providers** según quien lo usa:
  - **XletBoundProviders**: Cada XLET tiene su copia.
  - **SystemBoundProviders**: Una copia para todo el STB.
- Providers Soportados:
  - **SelectionProvider**: Traductor entre el Service Selection API / JMF y el protocolo propietario para comenzar la presentación, por ejemplo Switched Digital o RTSP (Real Time Streaming Protocol)
  - **SimpleSIProvider**: Traductor entre la System SI Database y formatos propietarios.
  - **SIManagerProvider**: Sobre-escribe el Java TV SI manager API para un Xlet.
  - **CryptographicServiceProviderProvider**. Ved MHP 1.1.3 “Advanced smart card usage”
  - **InteractionChannelTransportProvider**: Traductor entre el mecanismo de MHP 1.1 de “return channel download” y protocolos propietarios de download de ficheros.

## MHP 1.2

- **Operator Controlled MHP Terminals**

- MHP se centra más en productos más “comerciales” que en productos del operador de red.
- El uso de MHP en mercados donde el operador necesita de control ha implicado la inclusión de ciertas extensiones a MHP para satisfacer estas necesidades.
- OCAP ha estandarizado dichas extensiones: **Monitor Applications**.
- MHP incluye lo que denomina “**Priviledged Applications**” para ofrecer más control a los operadores:
  - El diseño está basado en el de monitor application de OCAP.
  - Requiere de una autenticación especial, lo cual puede ser resuelto mediante System Software Download.

## MHP 1.2

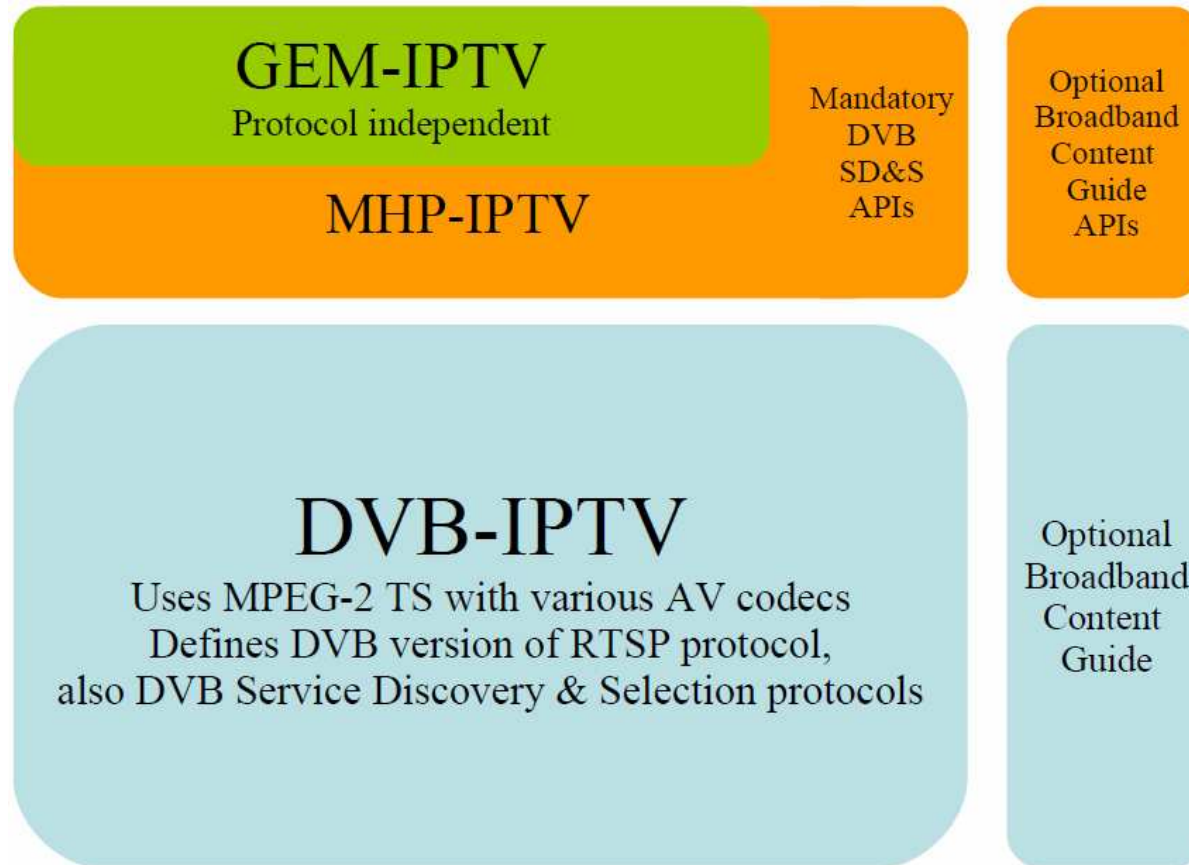
- **Operator Controlled MHP Terminals**

- Veamos qué pueden hacer las **Priviledged Applications**:

- Adoptado de OCAP:
  - Añadir / Quitar APPs de la applications database
  - Crear / Gestionar Services & Service Contexts
  - Controlar los permisos dados a las apps
  - Aceptar o Rechazar Broadcast apps
  - Reboot del STB
  - Recepción de notificaciones de errores y de escasez de recursos
- De MHP se adopta:
  - Procedimiento de aprobación de peticiones de almacenamiento de apps definido en el API de MHP 1.1 Application Storage API
- No se adopta de OCAP
  - Gestión de recursos
  - Enrutado de User-Events.
  - Otros: EAS, VBI, ..

## MHP / GEM for IPTV

Veamos un poco más en detalle en qué consiste



## MHP / GEM for IPTV

- **DVB-IPTV:** Protocolos Standard para IPTV incluyendo:
  - Broadcast / Multicast service discovery
  - Provisión de contenido Video & audio
  - Standard profile del RTSP(Real Time Streaming Protocol) protocol
  - Optional broadband content guide
- **MHP-IPTV:** Integra MHP con DVB-IPTV mediante:
  - Reutilización intensiva de los APIs existentes
  - Se añaden algunos nuevos APIs incluyendo extensiones específicas a algunos APIs de DVB-IPTV
  - Broadband content guide remains optional
- **GEM-IPTV**
  - Diseñado para redes que usan sistemas IPTV propietarios
  - Subconjunto de MHP-IPTV sin los protocolos DVB-IPTV

## GEM-IPTV en una red actual

GEM-IPTV

RTSP, IGMP & UDP protocol mapping

Protocol support for talking to  
Proprietary IPTV systems  
e.g. MSTV, OMP, Minerva etc

## APIs Comunes entre GEM-IPTV y MHP-IPTV

- **Existing Java TV APIs**

- El API de Service list funciona para IPTV broadcast (multicast) services
  - Lista/Lookup de IPTV broadcast services
  - Metadata de IPTV broadcast services
- Service selection API funciona para contenido IPTV
- Java Media Framework funciona para contenido IPTV

- **Nuevos APIs para IPTV**

- org.dvb.service añade soporte para receptores híbridos.

- **Contenido de tipo metadata bajo demanda no se incluye**

- El acceso a Metadata debe ser parte de las apps
- Presentación de contenido bajo demanda via Service Selection API o JMF

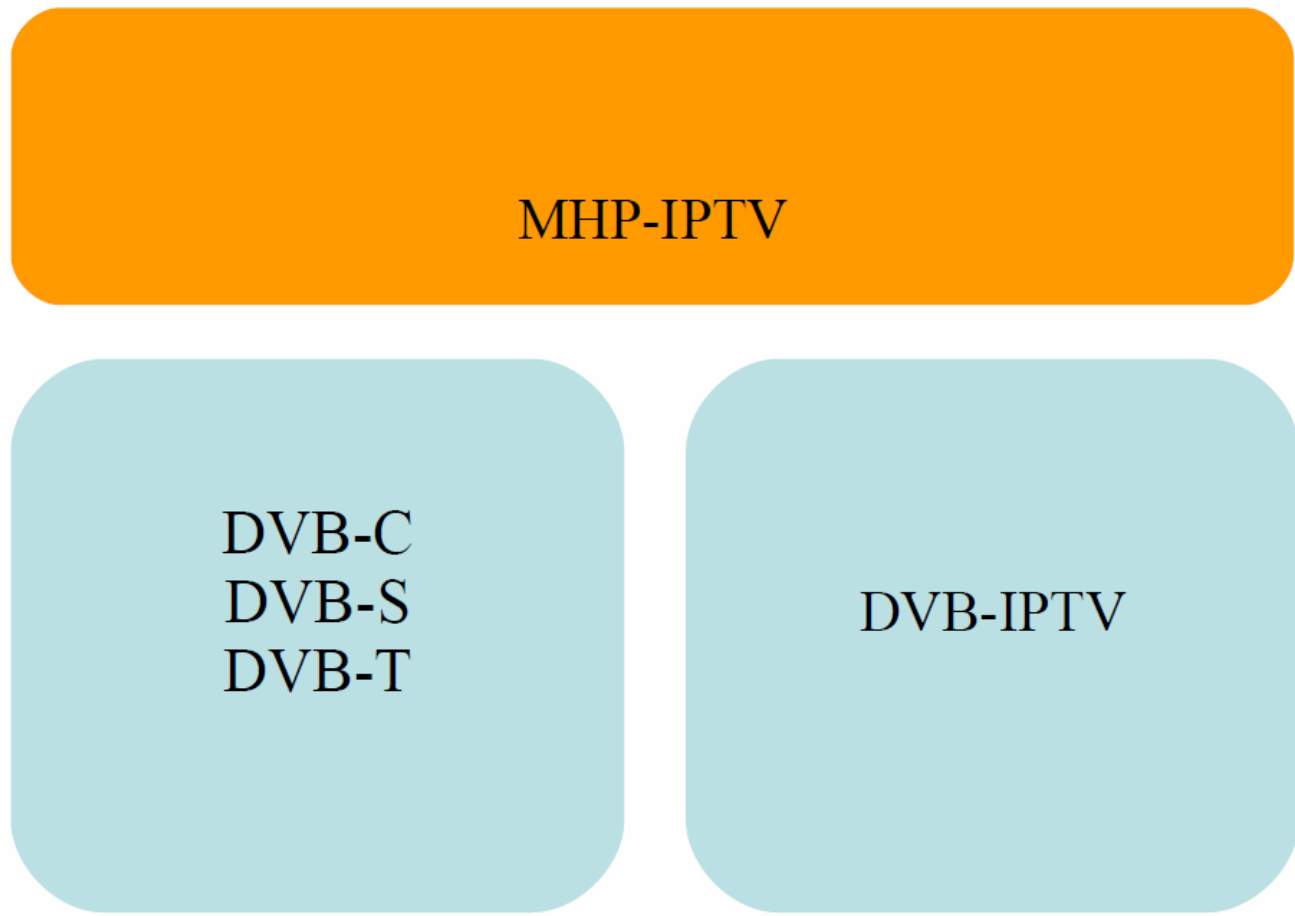
## Integración de GEM-IPTV con las DVB IPTV Specs

- **javax.tv.service**
  - Mapping a DVB-IPTV SD&S (Service Discovery & Selection) protocol
  - Mapping a DVB-IPTV BCG (Broadband Content Guide) protocol
- **Nuevos APIs**
  - org.dvb.service.sds amplía Java TV para obtener acceso detallado al SD&S protocol
  - org.dvb.tvanytime APIs give access to BCG (Broadband Content Guide)
    - Re-utilizado del MHP-PVR API
    - Ampliado para integrarlo con Java TV e IPTV
- **Extensions to SD&S to signal MHP applications**
  - Traducción del MHP signalling actual a XML

## Receptores Híbridos

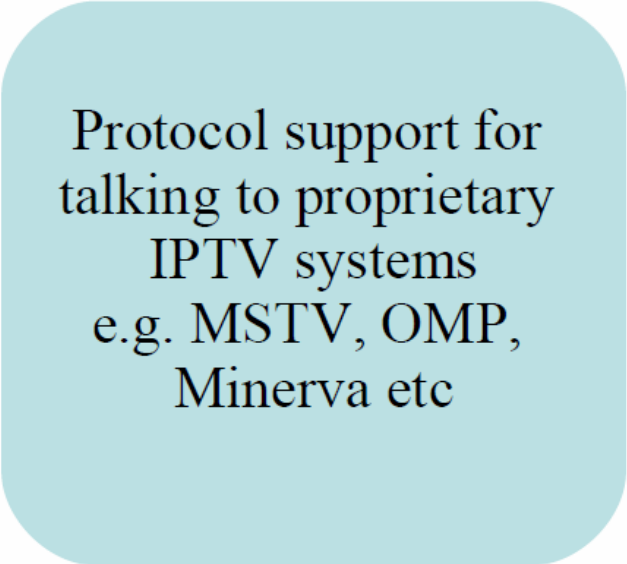
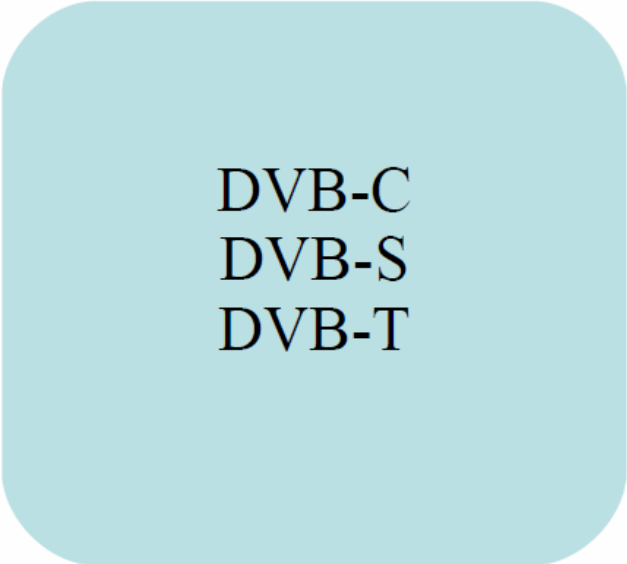
- Es Bastante habitual encontrar receptores IPTV con tuners clásicos:
  - DVB-T + IPTV
  - DVB-C + IPTV vía DOCSIS es posible
- La funcionalidad principal a completar es el concepto de Servicio independiente del Transporte en en Java TV
  - Los Services que aparecen en la lista de Services son independientes del Transport: Las Apps usan esos services sin preocuparse de cómo se reciben.
  - Aquellas apps a las que les importe como se recibe un Service siempre pueden transformar un `TransportIndependentService` en un `TransportDependentService`. Ved `org.dvb.service`
- 2 Tipos de STBs Híbridos soportados
  - Fully & partly standard

## Fully



tam1032r1-mhp-iptv-presentation.pdf

## Partly (1/2)



## Partly (2/2)

