



# Curso Multimedia Home Platform 1.1.2

## DSMCC I. Files & Directories

Qué es DSMCC

Leyendo Archivos

## Curso Multimedia Home Platform 1.1.2

Copyright 2008 © Enrique Pérez Gil

Licensed under the ***Creative Commons Attribution-Non-Commercial-No Derivative Works 3.0 Unported License***. You may not use this file except in compliance with the License. You may obtain a copy of the License at:

<http://creativecommons.org/licenses/by-nc-nd/3.0/legalcode>

This is a human-readable summary of the License applied:

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>)

**You are free to Share**, to copy, distribute and transmit the work **Under the following conditions:**

- **Attribution.** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Noncommercial.** You may not use this work for commercial purposes.
- **No Derivative Works.** You may not alter, transform, or build upon this work.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.

## Documentación a tener presente

### **ETSI TR 101 202 V1.2.1**

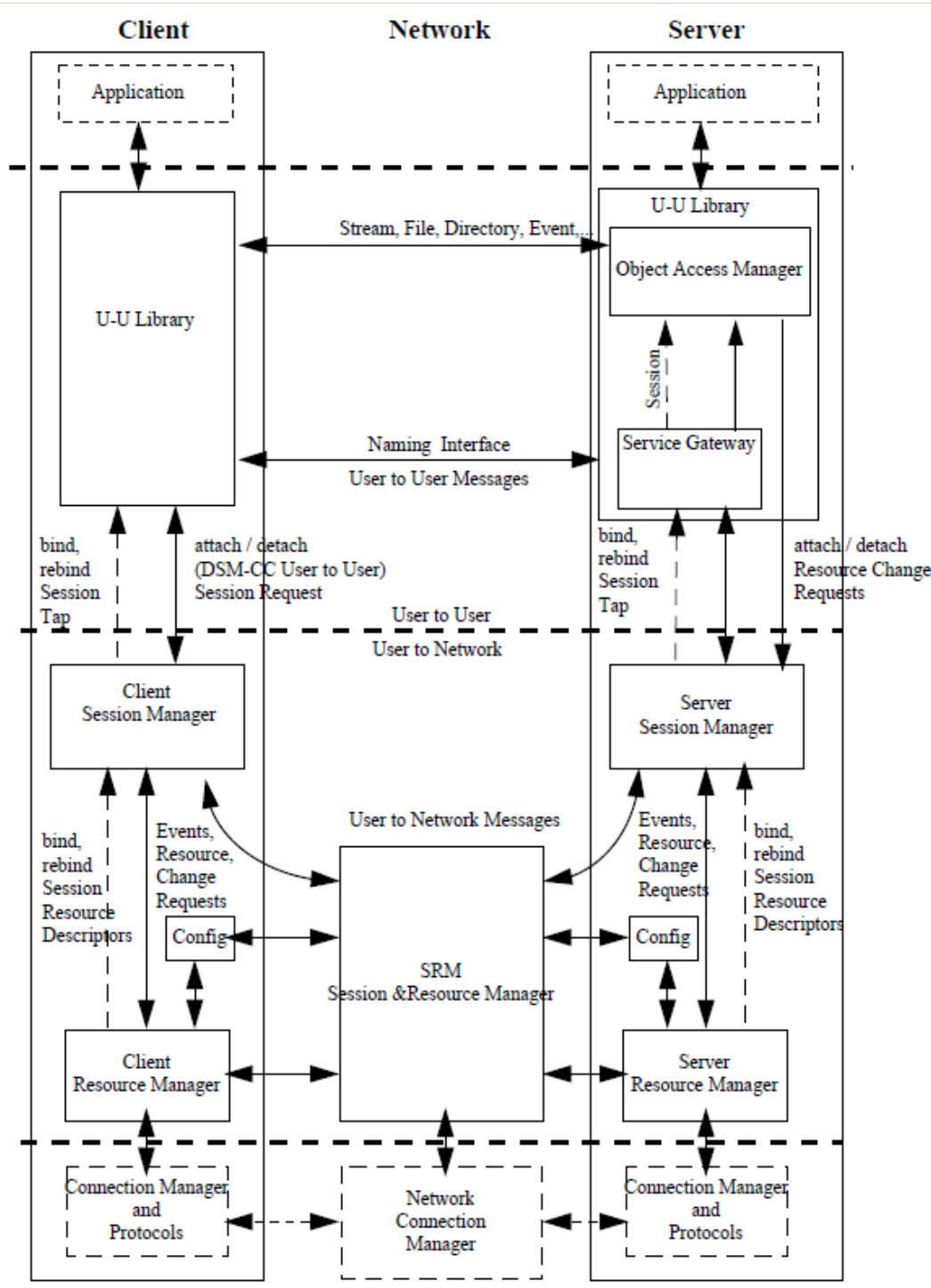
Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting

### **ISO-IEC 13818-6**

Extension for Digital Storage Media Command and Control

## Introducción. Recapitulemos

- ¿ **Qué es DSM-CC** ? : Digital Storage Media, Command and Control. Es una especificación complejísima que forma parte de MPEG-2 (parte 6) diseñada inicialmente para controlar contenido de tipo media sobre redes (VOD Servers, VTR...), pero que finalmente se amplió para proporcionar servicios de aplicaciones sobre redes heterogéneas.
- En el modelo DSM-CC, un Stream es proporcionado por un Server hacia un Client. Ambos, Server y Client se consideran Users de la red DSM-CC. DSM-CC define además lo que denomina “Session and Resource Manager” (SRM), lo cual proporciona una gestión (logica) centralizada de las sesiones y recursos de la red DSM-CC.
- Existen por tanto dos tipos de APIS/Mensajes dentro de DSM-CC, aquellos que sirven para hablar con la RED: **DSM-CC User-to-Network**, y otros, los que nos interesan, que permiten a los clientes y servidores “hablar” entre ellos y manipular **objetos** y contenido **media**: **DSM-CC User-to-User**.



## Introducción

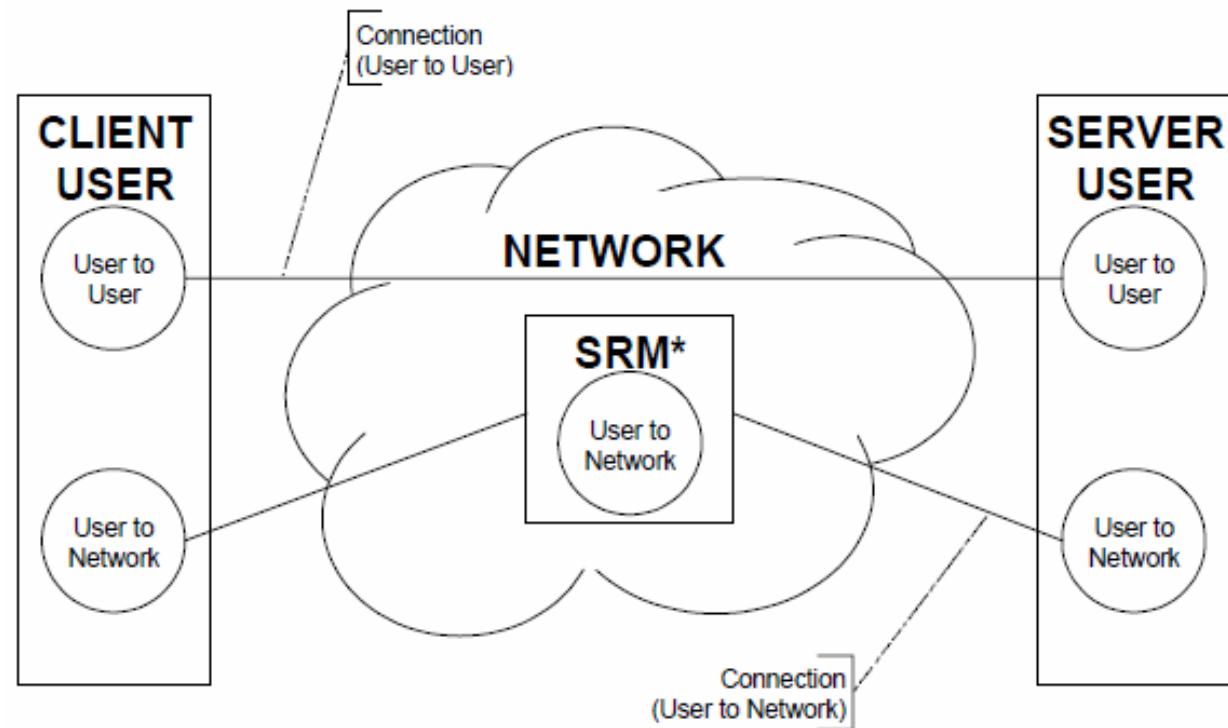
- En definitiva, como se ve en la figura, se ha definido un esquema con nuevos protocolos para poder construir servicios sobre datos MPEG-2 y estructuras de redes heterogéneas ( ATM, Asynchronous Serial Interfaces, IP Networks...)

ISO 13818-6



## Introducción

- Esquema simplificado de la arquitectura.



ISO 13818-6

## Introducción

- Funcionalidades que ofrece DSM-CC
  - Configuración de Red y establecimiento de Sesión
  - Control de Media Streams
  - **Sincronización con contenido Media**
  - **Acceso a ficheros** en redes tanto de un sentido (Broadcast) como de dos (IP)
  - Tecnologías de Data Broadcasting no basada en ficheros.
  - **Envío de Streams de Datos (Datagrams) en redes Broadcast.**
- Nos vamos a centrar sobre todo en:
  - **Acceso a Ficheros y Directorios**
  - **Stream Events**
  - **NPT**

## Introducción

- El Acceso a Ficheros en Broadcast se realiza mediante el protocolo **DSM-CC User-to-User Object Carousel**
- Básicamente consiste facilitar acceso a los Ficheros y Directorios que están en el servidor llevándolos al cliente (recordemos que estamos en Broadcast), de manera que para este su ubicación en cuanto su operación es “*transparente*”.
- El mecanismo se implementa mediante la publicación de los ficheros y directorios en la señal con una cierta frecuencia
- Esta frecuencia se puede establecer por ficheros, de manera que aquellos que se usan más se pueden publicar con una frecuencia mayor

## Introducción

- Un **Object Carousel** se apoya por debajo en el protocolo **DSM-CC Data Carousel**, que se apoya a su vez en las **DSMCC-Sections**, que añaden cierta lógica para poder manejar los mensajes DSI en la capa final **MPEG-Sections**
  - Interesante: Table B.1: DSM-CC Section Format, ETSI TR 101 202 V1.2.1
  - 4.7.6 MPEG-2 Sections, ETSI TR 101 202 V1.2.1

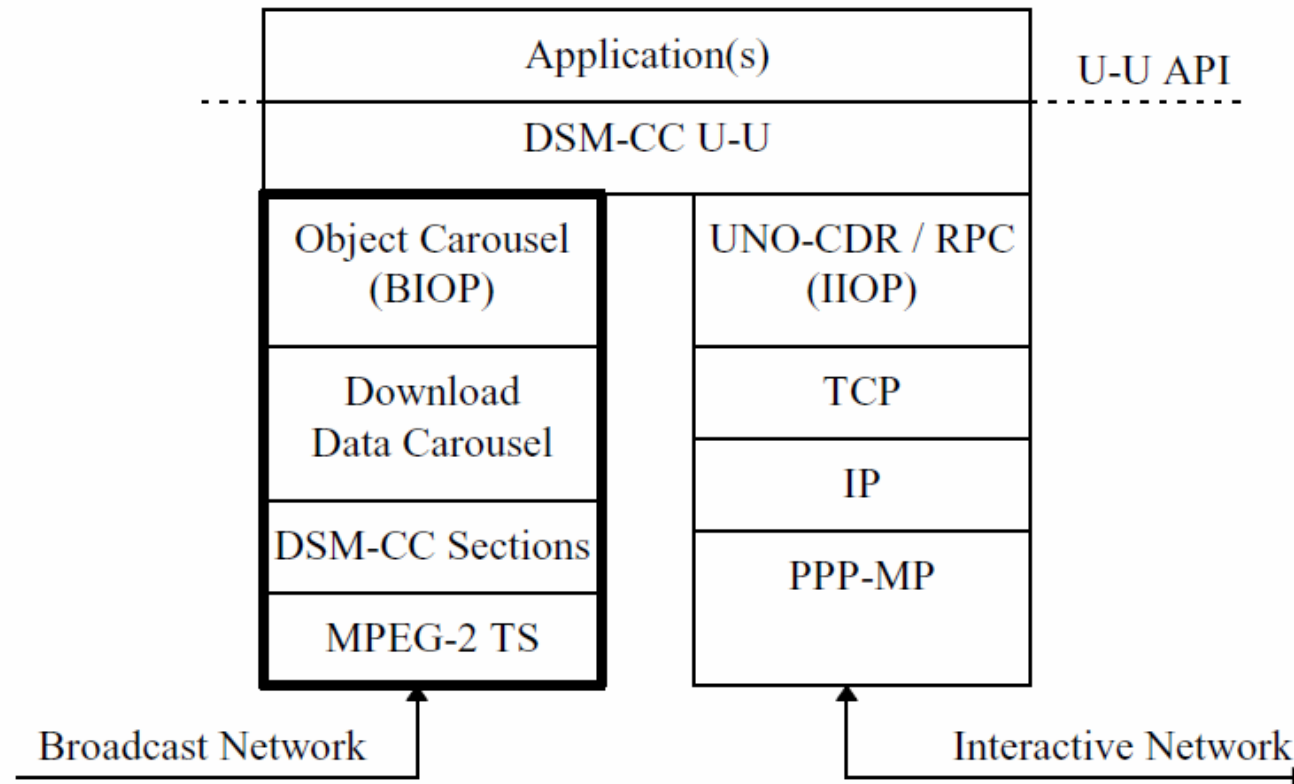
## DSM-CC Data Carousel

- No entiende de Ficheros o Directorios, sólo de **Módulos de datos**, y de los **bloques** que los componen.
- **Estos Módulos son los que se publican con una frecuencia**, en realidad un **Object Carousel** es una abstracción Lógica sobre los **Data Carousels**
- Las restricciones de tamaños en el envío de datos son:
  - Un módulo puede contener ficheros que sumen un tamaño máximo de **64KBytes**
  - Si un fichero es de más de 64KBytes NO se partirá en varios módulos sino que irá el solo en un módulo de su tamaño.



## Introducción

- Object Carousel Protocols



## Introducción

- **Object Carousel Transport Packets**

- Observad que a nivel de U-U se manejan los **BIOP Messages**. Vienen a ser como las Tablas de DVB SI, pero en el contexto de DSMCC

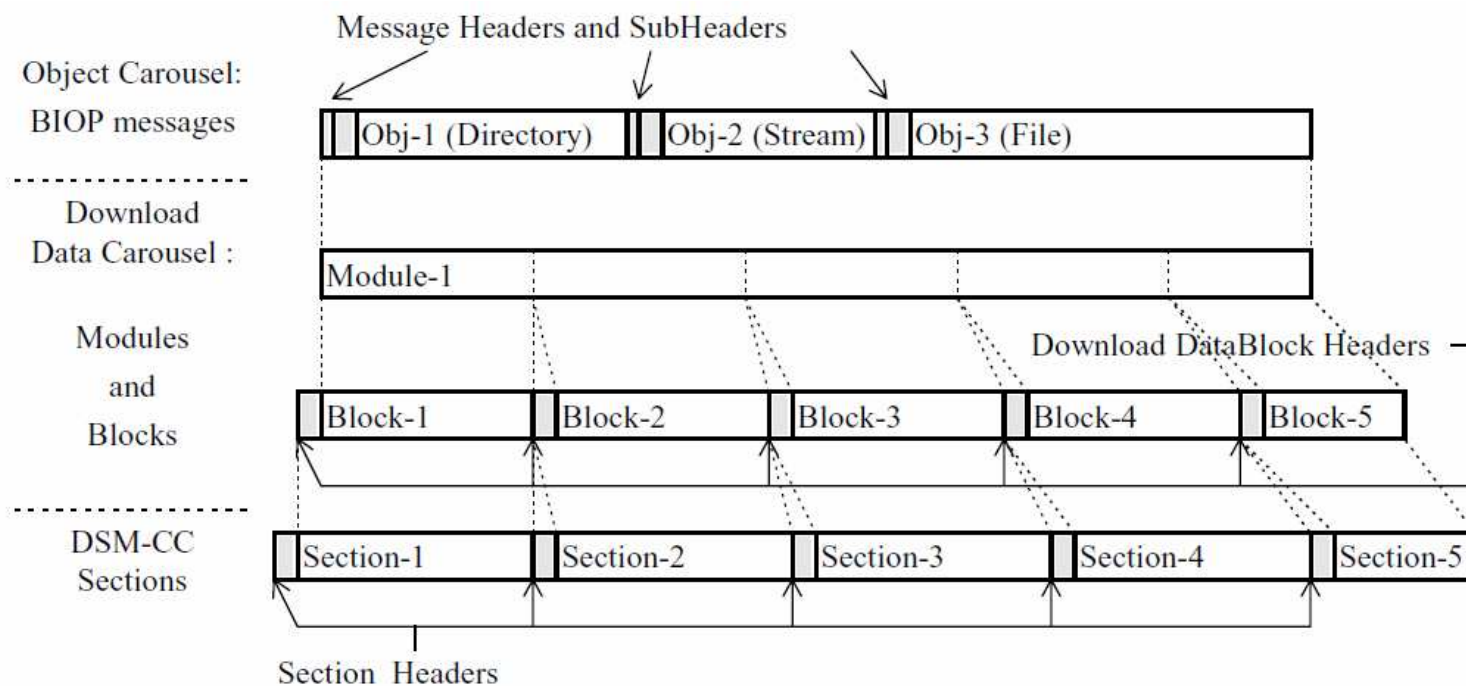


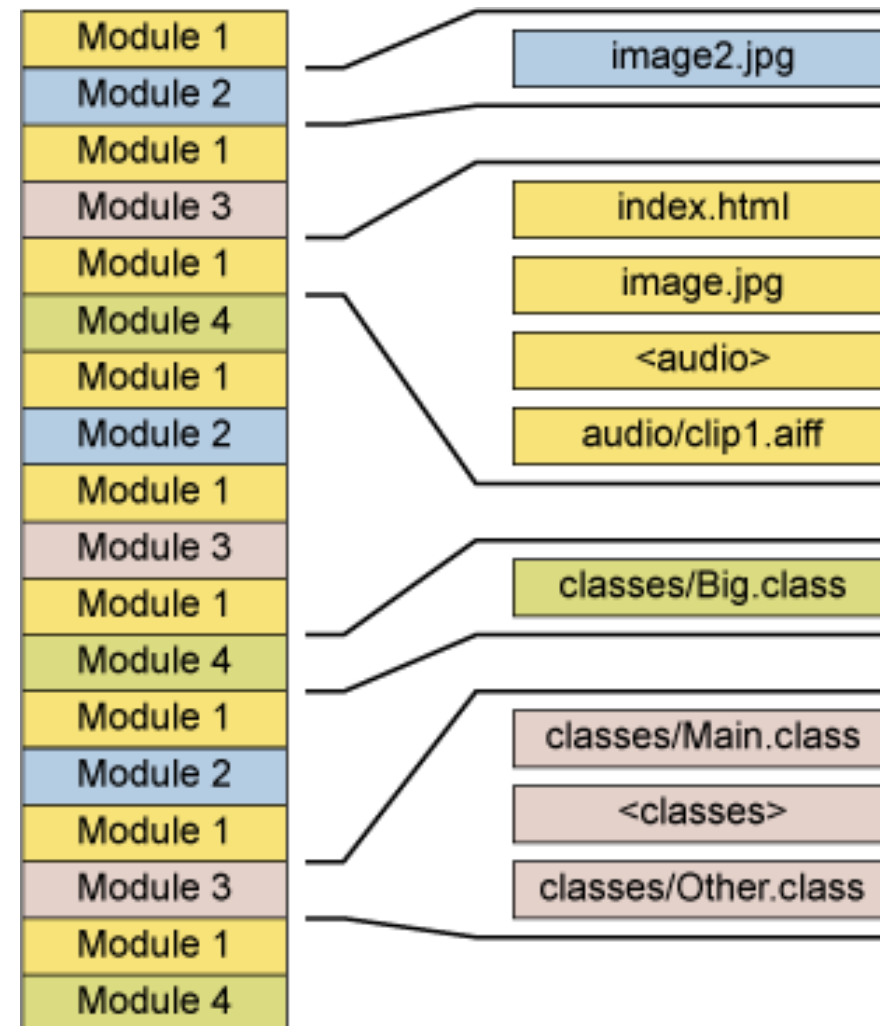
Figure 4.7: Encapsulation and fragmentation of BIOP Messages in Modules, Blocks, and MPEG-2 sections



## Introducción

- Un ejemplo de organización de ficheros en módulos

[www.interactivetvweb.org](http://www.interactivetvweb.org)



## Introducción

- Tened esto en mente: un Object Carousel te ofrece acceso a una serie de Objetos como si fuera un sistema estático, sin embargo por debajo es completamente dinámico.
- Y muy importante: en un Object Carousel pueden existir otros objetos que no tienen porqué ser Ficheros y Directorios, pero que igualmente desde el punto de vista que quien va a acceder a ellos, serán elementos “estáticos” que pueden estar o no en un momento dado, lo que ocurre es que la forma de llegar a ellos o de conocer su existencia en el Carousel será distinta a la de un Directorio.

## Introducción

- Ahora que ya sabemos lo que es un Object Carousel podemos decir que apoyándose en él, el protocolo **DSMCC-U-U** permite manejar **otros tipos de objetos**, no solo ficheros y directorios; veamos todos los tipos: (TR 101 202 4.7.2.3 Transmission of objects).

Dentro del grupo de Ficheros/Directorios

- **Ficheros**
- **Directorios** (referencia los ficheros que contiene)
- **Service Gateway**: Representa el directorio **RAIZ**. **Sólo puede haber uno por Object Carousel** -> en un Object Carousel sólo puede haber un Directorio de Ficheros

Otros tipos

- **Stream**: Es una lista de identificadores (denominados **TAPS**) que pueden apuntar a Streams de la misma Network (**BiopEsTap**) o a un DVB Service (**BiopProgramTap**). Con Stream Messages se pueden definir configuraciones de emisiones de programas: video + audio en un idioma determinado. Más adelante vemos otros tipos que puede incluir.
- **Stream Event**: Define **una lista de tipos de eventos** que pueden sucederse en el contexto de este Object Carousel; también en qué Stream se va a publicar **la sucesión de los eventos**, y además en qué Stream tendremos **la referencia de tiempo NPT**.

## Introducción

- Esquema de clases de los U-U Objects que usaremos. **Interesante: Stream Events hereda de Stream.**

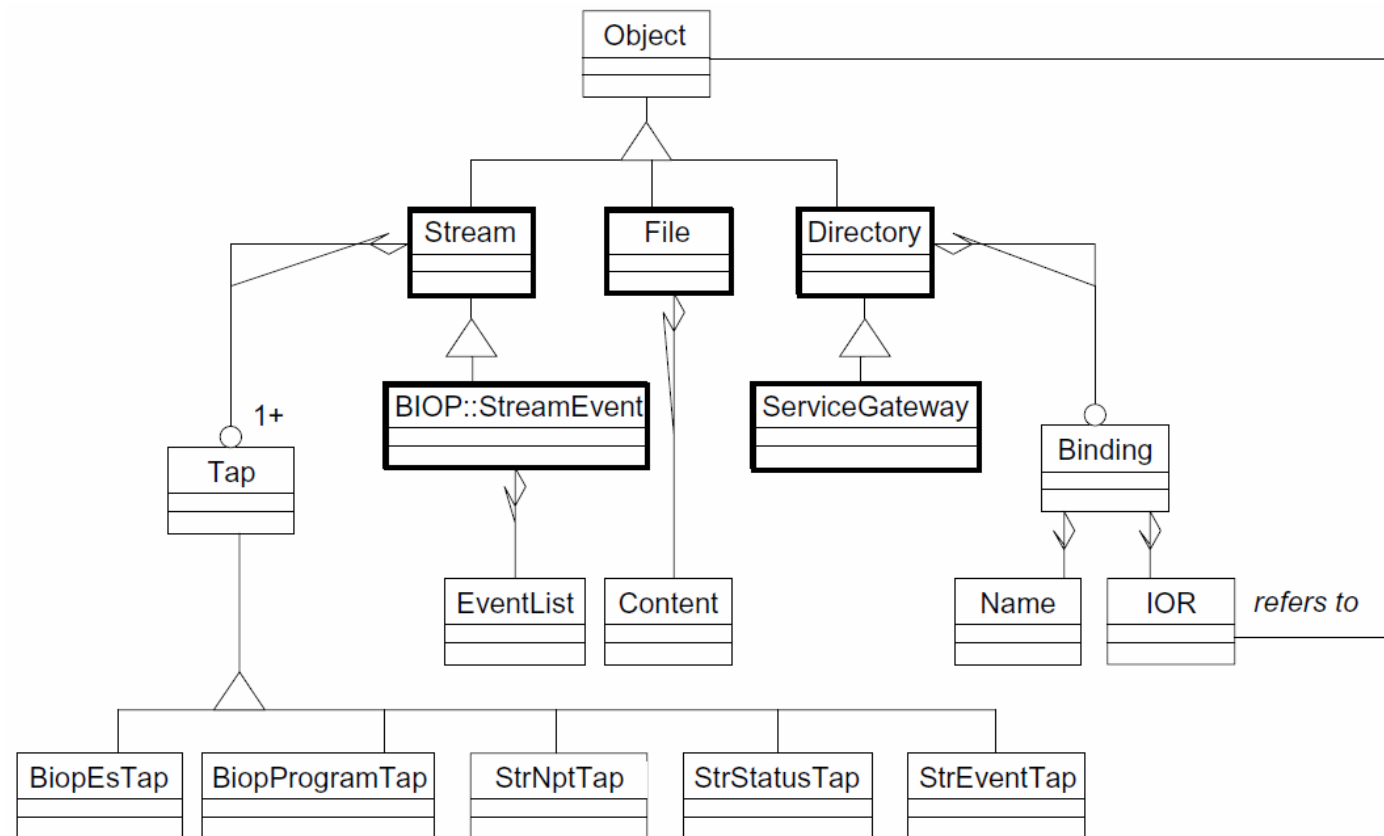
DSM::Directory

DSM::File

DSM::Stream

DSM::ServiceGateway

BIOP::StreamEvent



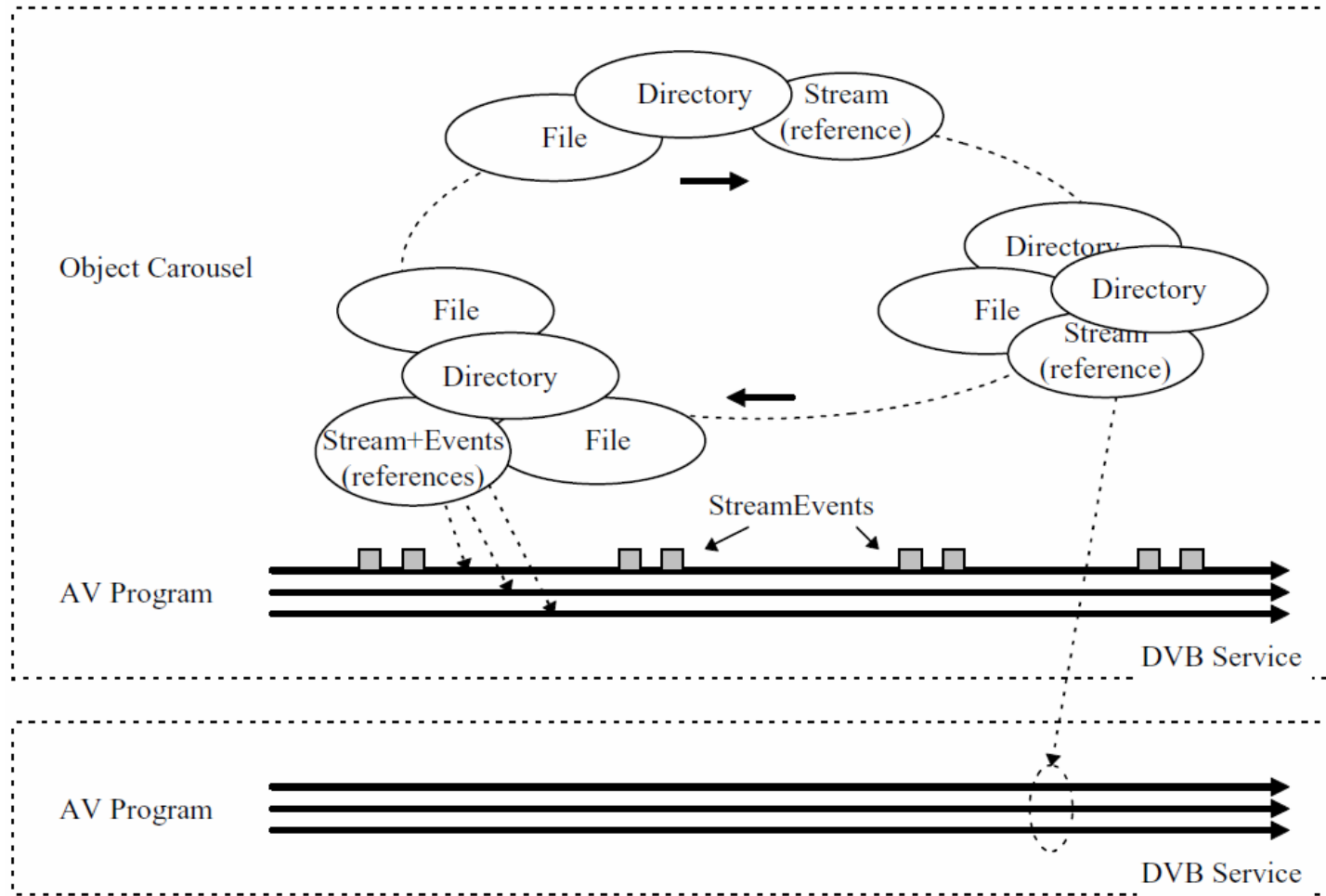
TR 101 202

## Introducción. Avance Stream Events

- Un Tipo de Evento no me vale de nada si no sé donde se indica la ejecución de un Evento ni en qué momento del tiempo debe producirse...
- **Stream Event** Message hereda de Stream, es decir, que además de la lista de Tipos de Eventos, cada uno con su **nombre**, características y su **EventId único en el contexto del Object Carousel**, dispone de las referencias(taps) a los Streams donde por un lado van a residir los **Stream Event Descriptors**, esto es, los “sucesos”, y por otro los **NPT Descriptors**, es decir, las referencias de Tiempo NPT: Normal Play Time.
- **NPT** es una referencia de tiempo que viaja como “**NPT reference descriptors**” en un **Stream de tipo 0x0C DSM-CC Descriptors**, y que permite al Broadcaster establecer una secuencia temporal válida para sus contenidos.
- **NPT es una ayuda que aporta DSMCC al Broadcaster.**
- Los Stream Event Descriptors también viajan en un **Stream de tipo 0x0C DSM-CC Descriptors**

## Introducción

- Los tipos gráficamente.



TR 101 202

## Introducción

- **DSM-CC-U-U** se apoya en CORBA para la descripción de los objetos que viajan en el **Object Carousel** (Ficheros, Directorios, etc.... ) los cuales son transmitidos como mensajes **BIOP, Broadcast Inter-ORB Protocol**; cada tipo de objeto tiene su tipo BIOP.
- **TAPS**: los **Stream Messages** usan **Taps** para referenciar streams que contienen:

TAP	VALOR	CONTENIDO EN EL ES PID
STR_NPT_USE	0X000B	<b>Stream NPT Descriptors</b>
STR_STATUS_AND_EVENT_USE	0X000C	Stream Mode + <b>Stream Event Descriptors</b>
STR_EVENT_USE	0X000D	<b>Stream Event Descriptors</b>
STR_STATUS_USE	0X000E	Stream Mode Descriptors
BIOP_ES_USE	0X0018	Elementary Stream (audio/video)
BIOP_PROGRAM_USE	0X0019	DVB Service

- Como vemos algunos de ellos son los usados por los **Stream Event**

## Introducción

- Al igual que ocurre con los Streams que llevan la información de la AIT, y que son de tipo **0x05** (private sections), es necesario igualmente especificar en la **PMT** los streams de tipo DSM-CC que forman parte del servicio
- Los tipos de Stream utilizados para DSM-CC son los siguientes:
  - **0x0A**: Multiprotocol Encapsulation: **Datos IP** (Los vimos en Protocols)
  - **0x0B**: Object o Data Carousels (Ficheros, Event Objects...): **Object Carousel y Data Carousel Messages**
  - **0x0C**: DSM-CC Descriptors, para **Stream Event Descriptors**, los “eventos” en sí, no los tipos, y **NPT Descriptors**: la referencia de tiempo. En detalle en el capítulo DSMCC II.
  - **0x0D**: es necesario abrirlo para conocer el contenido

## Un caso similar al de la AIT.

- ¿ Cómo sabemos si un stream de tipo 0x0B es un Object Carousel o un Data Carousel ? Al igual que cuando en la PMT comentábamos que la forma de saber si el Stream contenía la AIT era mediante la existencia de un **application\_signalling\_descriptor**, en este caso ocurre lo mismo: esperamos un **data\_broadcast\_id\_descriptor** (código 0x66)
- En la PMT nos encontramos la siguiente información respecto al Stream referenciado.
  - El tipo de Stream es el **0x0B**.
  - Se incluye un descriptor de tipo: **data\_broadcast\_id\_descriptor** (tipo 0x66) que contiene la siguiente información:
    - Si el valor del campo `data_broadcast_id` es 0x0006 hablamos de un Data Carousel. Si es 0x0007 es un Object Carousel (Ver Capítulo Protocols)
- Veamos a continuación el esquema de signalling para una APP en cuanto al transporte de la misma.

## [LINK: App Singlling] Transport Protocol Descriptor descriptor\_tag = 0x02 (y 6)

- Selector\_byte para **0x0001 Object Carousel** (Broadcast):
- **remote\_connection**: Cuando vale “1” se está indicando que el Object Carousel no lo proporciona el mismo Service donde se está publicando la AIT. Por eso en el IF se da la información necesaria.
- **component\_tag**: es el tag con el que ha sido “taggeado” (stream\_identifier\_descriptor) el stream donde reside el Object Carousel, **el cual habrá de publicar un DSMCC-DSI Message** (capa de mensajes dentro de los cuales van los Objects) **donde se describe el ServiceGateway**: el directorio raíz del Object Carousel, un objeto específico, obligatorio y único para un Object Carousel.

Table 93: Syntax of selector bytes for OC transport

Syntax	Bits	Mnemonic
remote_connection	1	bslbf
reserved_future_use	7	bslbf
if( remote_connection == "1" ) {		
original_network_id	16	uimsbf
transport_stream_id	16	uimsbf
service_id	16	uimsbf
}		
component_tag	8	uimsbf

## Carousel Identifier Descriptor

- Es un descriptor OPCIONAL. En el primer bucle de la **PMT** nos podemos encontrar un **carousel\_identifier\_descriptor** (ETSI TR 101 202 V1.2.1), lo cual nos ofrece una referencia global a un Object Carousel permitiendo al sistema conocer de su existencia con rapidez y dando la opción a adelantar la carga del mismo, sin embargo DVB recomienda que este descriptor se incluya en el ES Loop de la PMT, de manera que se puedan definir más de un OC y además nos identifique directamente en qué ES residen.

Table 4.17: carousel\_identifier\_descriptor

Syntax	bits	Type	Value	Comment
carousel_identifier_descriptor () {				
descriptor_tag	8	uimsbf	0x13	
descriptor_length	8	uimsbf	*	
carousel_id	32	uimsbf	+	
FormatId	8	uimsbf		Registered Identifier of the FormatSpecifier
FormatSpecifier(){				
FormatSpecifier_byte	8	uimsbf		(see table 4.17a) N2 bytes
}				
for ( i=0;i<N1;i++){				
private_data_byte	8	uimsbf		
}				
}				

## Accediendo a Archivos

- Tal y como hemos visto DSM-CC nos ofrece la posibilidad de hacer broadcast de estructuras de directorios.
- Las aplicaciones MHP podrán acceder a la estructura de directorios que la componen: el Object Carousel.
- ¿ Cómo ? Mediante la siguiente clase, que por cierto **hereda de java.io.File**  
**org.dvb.dsmcc.DSMCCObject**
- El API se encuentra en el paquete **org.dvb.dsmcc**

## Accediendo a Archivos

- En realidad la JVM de nuestra aplicación (en teoría una por App) toma como directorio de trabajo aquel en el que reside nuestra app.
- Sin embargo no es lo mismo acceder a un File en un PC que en un sistema en el que los ficheros están disponibles durante ciertos espacios de tiempo, aunque esto sea transparente al usuario.
- Aspecto FUNDAMENTAL: Es necesario usar un modelo asíncrono de acceso a recursos.

## Accediendo a Archivos

- Veamos el API de DSMCC. Creando un DSMCCObject.

- **org.dvb.dsmcc.DSMCCObject** extends **java.io.File**

- Representa un objeto en el Object Carousel.
- Se contruye con paths absolutos o relativos:

Relativo: `new DSMCCObject("a/b/c.txt");`

Absoluto: `new DSMCCObject("/r/f.txt");`

- Si lo haces con absolutos debes conocer el directorio Base de tu APP, **dato que ofrece la System property: user.dir** (ver APP Signalling)
- Si tu clase se llama: `com.tunombre.xlet`, el directorio **com** se crea bajo el base:

`/dir_base/com`

De tal forma que para referirte a un fichero que está en el base: `/dir_base/datos.tx` harás:

`new DSMCCObject("/dir_base/datos.txt");` ó `new DSMCCObject("datos.txt");`

- Es como un File, de manera que cuando nos estamos refiriendo a un Fichero deberemos usar las Clases `java.io.FileInputStream` o `java.io.RandomAccessFile` y cuando sea un directorio las llamadas apropiadas en `java.io.File`
- Cuando hacemos `new DSMCCObject(".")` estamos abriendo el directorio Base de la aplicación, esto es, donde residirían las clases cuyo package es "".

## Accediendo a Archivos

- El API de DSMCC. Carga Asíncrona

- public void **asynchronousLoad** (AsynchronousLoadingEventListener l)

- Se le indica al DSMCCObject que cargue el fichero asíncronamente.
- Cuando tenemos una cantidad importante de ficheros que cargar o bien el tamaño es grande, hemos de usar al acceso asíncrono que ofrece DSMCCObject. Que permite cargar el fichero de forma asíncrona en Caché!!!

```
public interface AsynchronousLoadingEventListener extends java.util.EventListener {  
    public void receiveEvent(AsynchronousLoadingEvent e);  
}
```

- En realidad el evento que se recibe es uno de los siguientes:

**SuccessEvent**, InvalidFormatEvent, InvalidPathNameEvent, MPEGDeliveryErrorEvent, ServerDeliveryErrorEvent, ServiceXFRErrorEvent, NotEntitledEvent, LoadingAbortedEvent, InsufficientResourcesEvent

- La carga de ficheros en modo asíncrono (en general todas las operaciones en modo asíncrono) tiene implicaciones importantes en cuanto al diseño de nuestro User Interface

## Accediendo a Archivos

- El API de DSMCC
  - public boolean **isLoaded()**
    - Si el objeto está cargado en caché.
  - public void **synchronousLoad()**
    - El método se bloquea hasta que el contenido se ha cargado en Caché. Ojo: se puede abortar su carga si desde otro Thread se llama a **abort()**.
    - No es necesario llamarlo para leer un fichero, se puede usar el API java.io directamente.
  - public boolean **isStream()**
    - True si el DSMCCObject es un DSMCC Stream Object. *DSMCCObject sirve para cargar otros tipos de objetos además de Files/Directories.*
  - public boolean **isStreamEvent()**
    - True si el DSMCCObject es un DSMCC StreamEvent Object.
  - public boolean **isObjectKindKnown()**
    - True si se conoce el tipo del objeto

## Accediendo a Archivos

- El API de DSMCC
  - public java.net.URL **getURL()**
    - URL del DSMCCObject. **Ojo:** null si el directory padre no está cargado.
  - public void **unload()**
    - Indica a MHP que el objeto no está siendo usado por otros elementos y que podría prescindir de los recursos cargados.
  - public void **abort()**
    - Nos permite abortar cualquier carga en curso (desde un Thread distinto cuando sync, 😊 ) : tanto las realizadas por asynchronousLoad como por synchronousLoad
  - public void **loadDirectoryEntry**(AsynchronousLoadingEventListener l)
    - Nos permite cargar de forma asíncrona la información de ficheros/directorios contenidos en un Directorio.

## Accediendo a Archivos

- El API de DSMCC. Adelantando la carga de contenido...
  - public **static** boolean **prefetch**(String path, byte priority)
    - Indica al middle que el contenido del path absoluto pasado interesa que se vaya cargando. El middle puede “pasar de nosotros”.
    - Cuanto mayor sea priority más importante que lo cargue.
    - Te devuelve true si la plataforma soporta pre-fetching
  - public **static** boolean **prefetch**(org.dvb.dsmcc.DSMCCObject dir, java.lang.String path, byte priority)
    - Variante del anterior para solicitar el pre-fetch del objeto referido con el path relativo al dir.
    - Te devuelve true si la plataforma soporta pre-fetching

## Accediendo a Archivos

- El API de DSMCC. Gestión de la Caché
  - public void **setRetrievalMode** (int mode)
    - Es posible indicarle a un DSMCCObject de donde debe obtenerse su contenido. Se ofrecen 3 opciones como parámetro:

*DSMCCObject.FROM\_CACHE\_OR\_STREAM: busca en caché y si no está en Stream*

*DSMCCObject.FROM\_STREAM\_ONLY: busca siempre en Stream*

*DSMCCObject.FROM\_CACHE: busca sólo en caché. Si no está se lanzará una excepción MPEGDeliveryErrorEvent*

## Accediendo a Archivos

- El API de DSMCC
  - public void **addObjectChangeListener**(ObjectChangeListener listener)
    - ¿ Y si el fichero ha sido cambiado por el Broadcaster? Normalmente esto se especifica cambiando la versión del mismo (a nivel de identificador de recurso dentro del Stream)

```
public interface ObjectChangeListener extends java.util.EventListener {  
    public void receiveObjectChangeEvent(ObjectChangeEvent e);  
}
```

```
public class ObjectChangeEvent extends java.util.EventObject {  
    public int getNewVersionNumber()  
    public Object getSource() // DSMCCObject  
}
```

## Accediendo a Archivos

- El API de DSMCC. Seguridad
  - `public X509Certificate[][] getSigners()`
    - Intentará validar la chain certificate del fichero. Los root NO tienen por qué ser conocidos del STB.
    - Ídem a `getSigners(false)`. Ver abajo.
  - `public X509Certificate[][] getSigners(boolean known_root )`
    - Ídem pero en el caso de `known_root=true` serán válidos aquellas chain certificate cuyo root sea conocido por el MHP Terminal

## Accediendo a Archivos

- El API de DSMCC. Seguridad
  - Para manejar Streams y Stream Events a partir de referencias de estos con DSMCCObject se usan las clases siguientes que pueden recibir en su constructor DSMCCObjects.

DSMCCStream

DSMCCStreamEvent

Hablaremos al respecto en DSMCC-2. Stream Events & NPT

## Ejercicios Bloque DSMCC1-1

## ServiceDomains

- Un Service Domain representa un Object Carousel, es decir, otro sistema de Archivos. En principio no existen restricciones para acceder a otros Object Carousels distintos del Object Carousel por defecto.
- ¿ cómo se accede a otro Object Carousel ? : Montando **org.dvb.dsmcc.ServiceDomain**
- La siguiente pregunta es: ¿ **Cómo se identifica un Object Carousel ?**
  - mediante Locators. En detalle en “DVB SI”. Existen varios:
    - org.davic.net.dvb.DvbLocator
    - org.davic.net.Locator
    - javax.tv.locator.Locator
  - Vienen a ser para MPEG lo que el objeto URL en Internet/File Systems. Sirven para identificar componentes dentro de Servicios. Así por ejemplo un Servicio se identifica como:  
`dvb://onID.tsID.sID`, donde onID: Original NetworkID, tsID: Transpor Stream ID y sID: Service ID.

## ServiceDomains

- Para montar un Object Carousel ServiceDomain ofrece 3 posibilidades:
  - public void **attach**(byte[] NSAPAddress)
    - poco usado. (NSAP Address de un ServiceDomain, definido en ISO/IEC 13818-6)
  - public void **attach**(org.davic.net.Locator l)
    - recibe un Locator que apunta a:
      - **Opción a:** un Service que tiene un solo Object Carousel, p.e.: dvb://12.12.34
      - **Opción b:** a un Elementary Stream **donde reside un Object Carousel:** dvb://32.34.45.34 (el cuarto parámetro es el PID que identifica un Stream o un TAG)
  - public void **attach**(org.davic.net.Locator aDVBSservice, int aCarouselId)
    - Se recibe un Locator de un servicio y un CarouselID, el correspondiente a un “Carousel Identifier Descriptor” que puede existir en la PMT asociado ya a un Stream, o bien (aunque no recomendable) en el primer Loop de la PMT

## ServiceDomains

- Ejemplo de montaje de un Object Carousel:

```
org.davic.net.dvb.DvbLocator dvbl = new org.davic.net.dvb.DvbLocator("dvb://22d4.a.105");
```

```
ServiceDomain sd = new ServiceDomain();
```

```
sd.attach(sd);
```

```
DSMCCObject dsmccObj = new DSMCCObject(sd.getMountPoint(), path);
```

- Nótese que **NO** se especifica donde se monta. Así se evitan problemas. Para acceder se usa getMountPoint()

## Ejercicios Bloque DSMCC1-2

<b>ISO/IEC 13818-1</b>	Part 1. Elementary Streams transport definition
<b>ISO/IEC 13818-6</b>	Part 6. Extensions for DSM-CC. Digital Storage Media Command and Control
<b>ETSI EN 300 468</b>	Digital Video Broadcasting (DVB);Specification for Service Information (SI) in DVB systems
<b>ETSI EN 301 192</b>	DVB specification for data broadcasting
<b>ETSI TR 101 202</b>	Implementation Guidelines for Data broadcasting
<b>ETSI TR 101 162</b>	Digital broadcasting systems for television, sound and data services; Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) systems
<b>ETSI TR 102 154</b>	Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in Contribution and Primary Dist
<b>ETSI TR 101 211</b>	Guidelines on implementation and usage of Service Information (SI)
<b>ETSI TR 101 200</b>	Digital Video Broadcasting (DVB); A guideline for the use of DVB specifications and standards
<b>DAVIC</b>	Digital Audio Visual Council. davic 1.4.1
<b>HAVI</b>	Specification of the Home Audio/Video Interoperability (HAVi) Architecture
<b>Interactivetvweb</b>	<a href="http://www.interactivetvweb.org/">http://www.interactivetvweb.org/</a>
<b>Wikipedia DSMCC</b>	<a href="http://en.wikipedia.org/wiki/DSM-CC">http://en.wikipedia.org/wiki/DSM-CC</a>
<b>MHP 1.1.2</b>	Multimedia Home Platform, A068r1 & tam668r23_11xdraft_20061115
<b>MHP 1.1.3</b>	Multimedia Home Platform, A068r3
<b>CDC 1.1</b>	Connected Device Configuration (CDC) 1.1 (JSR=218).
<b>PBP 1.1</b>	Personal Basis Profile 1.1 (JSR 217)
<b>MHP.org</b>	<a href="http://www.mhp.org">www.mhp.org</a>
<b>INTRO MHP 1.1.3</b>	tam1032r1-mhp-iptv-presentation