

Tecnologías y actividades  
de estandarización para la  
interconexión de *Home Networks*

Anexo C:

Descripción de tecnologías para *middleware*, *software*  
y pasarelas residenciales

---

*Alcatel* para Fundación AUNA

---

# Tecnologías y actividades de estandarización para la interconexión de *Home Networks*

1	<b>UPNP: <i>Universal Plug and Play</i></b>	3
	1.1. Descripción técnica general	3
	1.2. Presencia en el mercado	6
2	<b>JINI</b>	7
	2.1. Descripción técnica general	7
	2.2. Presencia en el mercado	9
3	<b>HAVI: (<i>Home Audio/Video Interoperativity</i>)</b>	10
	3.1. Descripción técnica	10
	3.2. Presencia en el mercado	11
4	<b>OSGI (<i>Open Services Gateway Initiative</i>)</b>	13
	4.1. Características principales.	13
	4.2. Descripción básica del funcionamiento	14
	4.3. Presencia en el mercado	15
5	<b>MHP (<i>Multimedia Home Platform</i>)</b>	16
	5.1. Descripción técnica	16
	5.2. Consideraciones de mercado	19

# 1. UPnP (UNIVERSAL PLUG AND PLAY)

*Universal Plug&Play* (UPnP) es una arquitectura SW abierta y distribuida que permite a las aplicaciones de los dispositivos conectados a una red intercambiar información y datos de forma sencilla y transparente para el usuario final, sin necesidad de que este tenga que ser un experto en la configuración de redes, dispositivos o sistemas operativos. Esta arquitectura SW está por encima de protocolos como el TCP, el UDP, el IP, etc. y es independiente de estos.

El UPnP se encarga de todos los procesos necesarios para que un dispositivo u ordenador conectado a una red pueda intercambiar información con el resto. El UPnP ha sido diseñado de forma que sea independiente del fabricante, sistema operativo, del lenguaje de programación de cada dispositivo u ordenador y del medio físico usado para implementar la red.

Este protocolo es capaz de descubrir cuándo se conecta un nuevo equipo o dispositivo a la red, asignándole una dirección IP, un nombre lógico, informando a los demás de sus funciones y capacidad de procesamiento, e informarle, a su vez, de las funciones y prestaciones de los demás. De esta forma, el usuario no tiene que preocuparse de configurar la red ni de perder el tiempo instalando *drivers* o controladores de dispositivos, el UPnP se encarga todos estos procesos cada vez que se conecta o se desconecta un equipo; y además, optimiza en todo momento la configuración de los equipos.

## 1.1. DESCRIPCIÓN TÉCNICA GENERAL

### 1.1.1. Componentes de una red UPnP

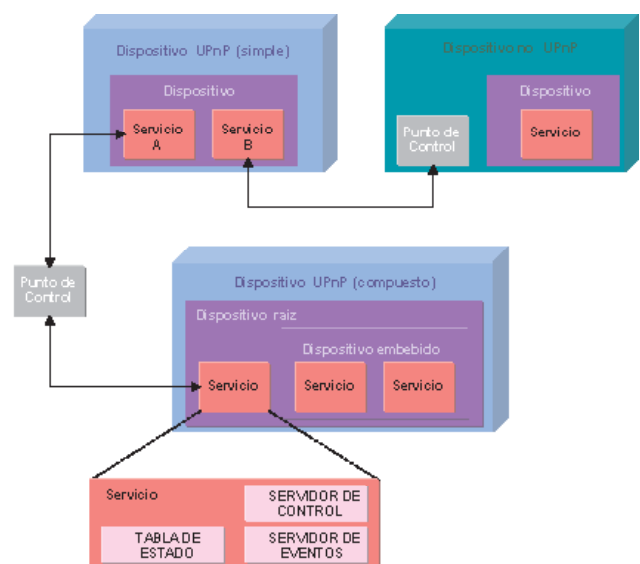
Los bloques de construcción básicos de una red UPnP son los dispositivos, los servicios y los puntos de control, cuya interrelación se muestra en la Figura C1, y que a continuación se describen:

- Dispositivos

Un dispositivo UPnP es un contenedor de servicios y/o de otros dispositivos. Por ejemplo, una videgrabadora puede consistir en un servicio de transporte de cinta, un servicio de afinación y un servicio de reloj. Mientras que un TV/videgrabadora consistiría no sólo en esos servicios y alguno más, sino también en otros dos dispositivos integrados.

Diferentes categorías de dispositivos UPnP se asociarán con diferentes conjuntos de servicios y dispositivos integrados. Por ejemplo, los servicios de una videgrabadora serán diferentes a los de una impresora. Como consecuencia, los diferentes grupos de trabajo se estandarizarán en el conjunto de servicios que proporcionará un tipo de servicio en particular. Toda esta información se captura en un documento de descripción de dispositivo XML que el dispositivo debe alojar y que incluye las propiedades asociadas con el dispositivo.

Figura C1. Puntos de control, dispositivos y servicios de UPnP



- Servicios

La unidad de control más pequeña en una red UPnP es un servicio. Un servicio expone acciones y modela su estado con variables de estado. Por ejemplo, un servicio de reloj se puede modelar para tener una variable de estado, hora actual, que define el estado del reloj y dos acciones, definir hora y obtener hora, que le permiten controlar el servicio.

Similar a la descripción del dispositivo, esta información es parte de una descripción de servicio XML estandarizada por el foro UPnP.

Además de la tabla de estado, un servicio en un dispositivo UPnP contiene un servidor de control y un servidor de eventos. El servidor de control recibe solicitudes de acción (tales como establecer hora), las lleva a cabo, actualiza la tabla de estado y devuelve respuestas. El servidor de eventos publica eventos para suscriptores interesados en cualquier momento en que cambie el estado del servicio. Por ejemplo, el servicio de alarma contra incendios enviaría un evento a los suscriptores interesados cuando su estado cambia a “sonar”.

- Puntos de control

Un punto de control en una red UPnP es un controlador capaz de descubrir y controlar a otros dispositivos. Después del descubrimiento, un punto de control podría:

- Recuperar la descripción del dispositivo y obtener una lista de servicios asociados.
- Recuperar las descripciones de servicio para los servicios de interés.
- Invocar acciones para controlar el servicio.
- Suscribirse a la fuente de eventos del servicio. Siempre que cambie el estado del servicio, el servidor de eventos enviará un evento al punto de control.

### 1.1.2. Descripción general del protocolo UPnP

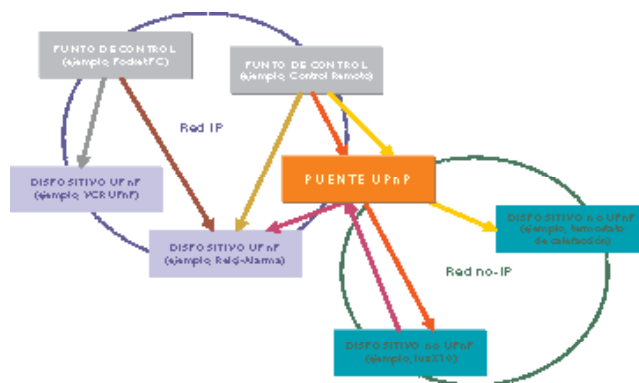
UPnP aprovecha el conjunto estándar del protocolo IP para que los medios de la red sigan siendo independientes. Los dispositivos en una red UPnP se pueden conectar utilizando cualquier medio de comunicación, incluyendo radio (RF, inalámbrica), línea telefónica, línea de energía, IrDA, Ethernet e IEEE 1394. En otras palabras, cualquier medio que se pueda utilizar para colocar dispositivos en red, UPnP puede habilitarlo. La única preocupación podría ser que los medios que se están utilizando no soporten el ancho de banda que se requiere para el uso que se pretende.

UPnP utiliza protocolos abiertos y estándares, tales como TCP/IP, HTTP y XML. Sin embargo, también se pueden utilizar otras tecnologías para colocar en red los dispositi-

tivos, incluyendo tecnologías de operación en red como HAVI, CeBus, LonWorks, EIB o X10.

Todas estas tecnologías pueden participar en la red UPnP a través de un “bridge” (puente) o proxy UPnP. Una red UPnP que contenga dispositivos en red a través de un “bridge”, se podría ver de manera similar a la Figura C2.

Figura C2. Una red UPnP utilizando un puente o “bridge”



Por otra parte, el UPnP no requiere de ninguna capa de software intermedia entre el sistema operativo y los protocolos de Internet sobre los que se sustenta la capacidad de relación de los dispositivos. El sistema operativo, eso sí, debe incorporar las interfaces de aplicación (APIs) para los distintos dispositivos. Así se convierte en una especificación de software no solo independiente del medio de transmisión, sino también del sistema operativo (aunque en la práctica, la mayoría de las aplicaciones actuales de UPnP están asociadas a Windows).

### 1.1.3. ¿Cómo funciona UPnP?

UPnP proporciona soporte para la comunicación entre los puntos de control y los dispositivos. Los medios de red, el conjunto de protocolo TCP/IP y HTTP proporcionan la conectividad de red básica y el direccionamiento necesario. Sobre estos protocolos abiertos basados en Internet, UPnP define un conjunto de servidores HTTP para manejar el descubrimiento, la descripción, el control, los eventos y la presentación.

#### Direccionamiento

El fundamento de la operación en red de UPnP es el conjunto del protocolo TCP/IP y la clave para este conjunto

es el direccionamiento. Cada dispositivo debe tener un cliente de Protocolo con configuración dinámica de *host* (DHCP) y buscar un servidor DHCP cuando el dispositivo se conecta por primera vez a la red. Si hay un servidor DHCP disponible, el dispositivo debe utilizar la dirección IP que se le asigna. Si no está disponible ningún servidor DHCP, el dispositivo debe utilizar la función de “Auto IP” para obtener una dirección.

Un dispositivo también puede implementar protocolos con niveles más altos fuera de UPnP que utilicen nombres fáciles de identificar para los dispositivos. En estos casos es necesario resolver nombres de *host* para obtener la dirección IP. Normalmente, para realizar esta función se utiliza DNS.

## Descubrimiento

Una vez que los dispositivos se conectan a la red y se direccionan adecuadamente, tiene lugar el descubrimiento, que es gestionado por el protocolo SSDP. Cuando un dispositivo se agrega a la red, SSDP permite al dispositivo anunciar sus servicios a los puntos de control de la red. Cuando un punto de control se agrega a la red, SSDP permite a ese punto de control buscar los dispositivos de interés en la red.

El intercambio fundamental en ambos casos es un mensaje de descubrimiento que contiene algunos aspectos específicos e importantes acerca del dispositivo o de alguno de sus servicios.

## Descripción

El siguiente paso en la operación en red UPnP es la descripción. Después de que un punto de control ha descubierto un dispositivo, el punto de control sigue conociendo muy poco acerca del mismo. Para que el punto de control aprenda más acerca del dispositivo y sus capacidades, o para interactuar con el dispositivo, el punto de control debe recuperar la descripción del dispositivo del sitio *web* proporcionado por el dispositivo en el mensaje de descubrimiento.

Los dispositivos pueden contener otros dispositivos y servicios lógicos. La descripción UPnP para un dispositivo se expresa en XML e incluye información del fabricante específica, que incluye el nombre y número de modelo,

número de serie, nombre del fabricante, direcciones de sitios web específicos del proveedor, etc. La descripción también incluye una lista de algunos dispositivos o servicios integrados, así como direcciones *web* para control, eventos y presentación.

## Control

Después de que un punto de control ha recuperado una descripción del dispositivo, el punto de control cuenta con los aspectos básicos para la gestión del mismo. Para aprender más sobre este servicio, un punto de control debe recuperar una descripción UPnP detallada para cada servicio. La descripción para un servicio también se expresa en XML e incluye una lista de los comandos, o acciones, a los que responde el servicio y los parámetros o argumentos para cada acción. La descripción de un servicio también incluye una lista de variables. Estas variables modelan el estado del servicio durante el tiempo de ejecución y se describen en términos, ya sea de tipo de datos, de rango y/o de características del evento.

## Eventos

Una descripción UPnP para un servicio incluye una lista de acciones a las cuales responde el servicio y una lista de variables que modelan el estado del servicio durante el tiempo de ejecución. El servicio publica actualizaciones cuando estas variables cambian y un punto de control puede suscribirse para recibir esta información.

El servicio publica actualizaciones al enviar mensajes de eventos. Los mensajes de eventos contienen los nombres de una o más variables de estado y el valor actual de esas variables. Estos mensajes también se expresan en XML y se formatean utilizando el protocolo GENA.

## Presentación

Si un dispositivo tiene una dirección *web* para presentación, entonces el punto de control puede recuperar una página a partir de esta dirección, cargar la página en un navegador y, dependiendo de las capacidades de la página, permitir al usuario controlar el dispositivo y/o ver el estado del mismo. El grado que cada uno de estos pueda lograr, depende de las capacidades específicas de la página y del dispositivo de presentación.

## 1.2. PRESENCIA EN EL MERCADO

Hay que destacar que el UPnP, que ha sido auspiciado por *Microsoft*, persigue los mismos objetivos que el Jini de *Sun Microsystems*. De hecho, puede considerarse como el “contraataque” de *Microsoft* contra el Jini de *Sun*, aunque este le lleva casi dos años de ventaja y experiencia en el mercado. Aún así, ya existen muchos productos que cumplen estas especificaciones (especialmente en el campo de la informática) y funcionan además con SCP.

UPnP sigue en fase de elaboración, y la gran dificultad de este protocolo es que necesita que haya acuerdos en cada subsector (impresoras, almacenamiento, audiovisual) para consensuar las interfaces de aplicaciones (APIs). De hecho, con el fin de implicar aún más el desarrollo de estos APIs, se ha integrado dentro del UPnP al grupo de trabajo HomeAPI (HAPI), que venía trabajando desde 1997 con el objetivo de especificación y desarrollo de un conjunto de servicios y interfaces de programación (*Application Program Interface - APIs*) orientados hacia la automatización y control de las viviendas.

Por otra parte, es de destacar que la arquitectura propuesta por UPnP presenta algunos inconvenientes que aún quedan por resolver y que pueden afectar a su aplicabilidad universal. Entre estos cabe citar:

- Es una arquitectura de *software* que permite únicamente la transmisión de datos entre dispositivos IP.
- Esta pensado para disponer del PC como elemento central.
- Aún presenta algunos problemas para comunicar dispositivos a través de *firewalls*.
- En esencia esta basado en SOAP, para utilización de dispositivos no IP tenemos que recurrir a SCP (*Simplify Control Protocol*) ■

## 2. JINI

El Jini es una tecnología, desarrollada por *Sun Microsystems*, que proporciona un mecanismo sencillo para que diversos dispositivos conectados a una red puedan colaborar y compartir recursos sin necesidad de que el usuario final tenga que planificar y configurar dicha red. En esta red de equipos, llamada “comunidad”, cada uno proporciona a los demás los servicios, controladores e interfaces necesarios para distribuirse de forma óptima la carga de trabajo o las tareas que deben realizar.

### 2.1. DESCRIPCIÓN TÉCNICA GENERAL

El objetivo de Jini es convertir la red en un sistema flexible y fácil de administrar, en el cual se puedan encontrar rápidamente los recursos disponibles de la red.

Un sistema Jini consiste en un sistema distribuido basado en la idea de grupo federativo de usuarios (comunidades) y de los recursos requeridos por ellos. Los recursos pueden ser implementados tanto por dispositivos *hardware* y *software* como por una combinación de ambos.

#### 2.1.1. Componentes

Las partes de un sistema Jini son:

Un conjunto de componentes que proporcionan una infraestructura de servicios federativos en un sistema distribuido.

Un modelo de programación que soporta y estimula la producción fiable de servicios distribuidos.

Los servicios que pueden ser parte de un sistema federativo Jini y que ofrecen funcionalidad a cualquiera de los miembros de la federación o “comunidad”.

Jini supone que la infraestructura de red sobre la que se monta tiene el ancho de banda necesario y es lo suficientemente fiable para funcionar, por lo que no aporta mecanismos para mejorar estos dos puntos. También se asume que los dispositivos Jini tienen capacidad de procesamiento y memoria suficientes.

#### 2.1.2. Servicios

El concepto de servicio es el más importante dentro de la arquitectura Jini. Un servicio es una entidad que puede ser usada por una persona, un programa u otro dispositivo. Un servicio puede ser de computación, de almacenamiento, un canal de comunicación con otro usuario, un filtro *software*, un dispositivo *hardware*, o cualquier usuario. La naturaleza dinámica de Jini permite que los servicios sean añadidos (eliminados), en cualquier instante, a (de) una federación, de acuerdo con las necesidades, demandas o cambios en los requisitos del grupo de trabajo que utilice la federación. Los servicios se comunican entre si utilizando el protocolo de servicio, el cual consiste en un conjunto de interfaces escritas en Java, que reposan sobre la tecnología de RMI (*Java Remote Method Invocation*).

Para saber los servicios que están disponibles se utiliza el Servicio de búsqueda (*Lookup Service*). Este “mapea” las interfaces que indican la funcionalidad de un servicio con el conjunto de objetos que implementan dicho servicio. El servicio de búsqueda se organiza de forma jerárquica.

#### 2.1.3. Protocolos

Cuando se quiere añadir un servicio a la tabla se utiliza el protocolo *discovery* y el protocolo *join*. El primero se encarga de buscar el servicio y el segundo de añadirlo.

Cuando se quiere utilizar el servicio se busca en la tabla de servicios (con el protocolo *look-up*) para ver si existe. En caso de encontrarlo el cliente se descarga el código de control de ese servicio, que puede ir desde una interfaz hasta la implementación completa del servicio.

Se incorporan también un mecanismo de transacciones, para agrupar varias operaciones en una sola, y un mecanismo de eventos.

#### 2.1.4. Leasing

El acceso a muchos de los servicios en un entorno Jini se basa en un sistema de “*leasing*”. Cada “*lease*” es una concesión que garantiza el acceso durante un periodo de

tiempo determinado. Este se negocia entre el proveedor del servicio y el cliente como parte del protocolo de negociación. La concesión puede ser exclusiva o no exclusiva.

El *leasing* exclusivo asegura que nadie más, dentro de la comunidad, va a utilizar el recurso durante el periodo establecido por el *leasing*, mientras que una concesión no-exclusiva permite a varios usuarios compartir el recurso.

### 2.1.5. Descripción del funcionamiento de Jini

Como ya se ha comentado, la esencia del funcionamiento de Jini reside en un trio de protocolos denominados "Discovery", "Join" y "Look-up".

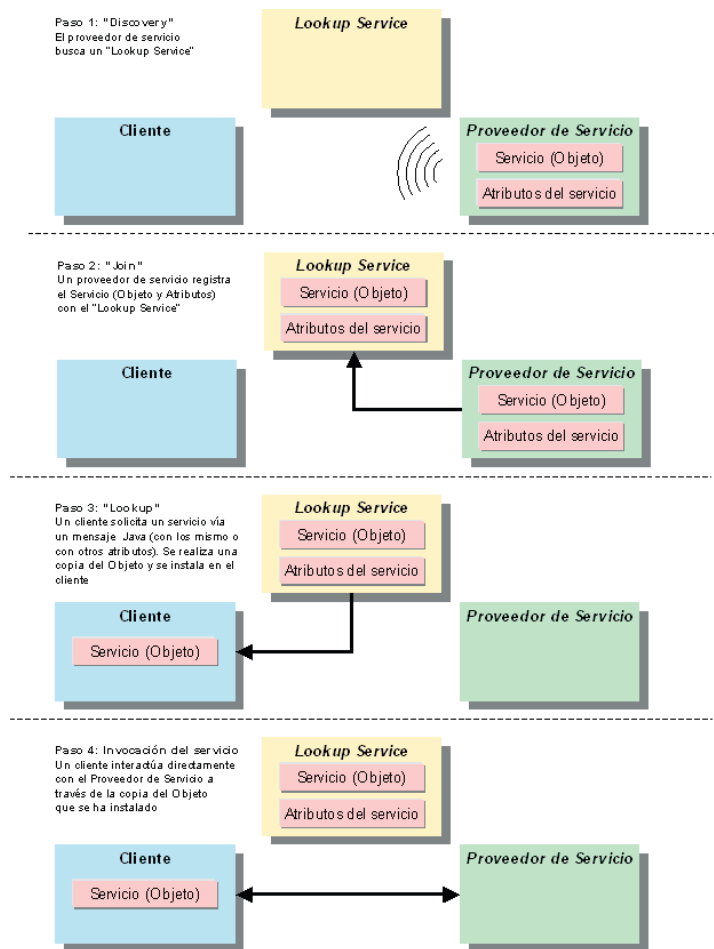
Cuando un dispositivo se inserta (*plug-in*) en la red, la pareja de protocolos "Discovery" y "Join" se encargan

de añadirlo a la comunidad o federación. El proceso de "Discovery" se produce cuando el nuevo servicio, proporcionado por el dispositivo insertado, busca en la tabla (o servicio) un "Lookup Service" al cual registrarse, mientras que el proceso "Join" se encarga de que el servicio determinado se registre en la tabla "Lookup Service" como disponible por la comunidad.

Cada vez que un cliente de la federación desee hacer uso de un determinado servicio, se lanzará un proceso de "Look-up" para comprobar si dicho servicio se encuentra en la tabla y, si es así, se cargará una copia del mismo en el cliente, estableciendo un enlace virtual directo con la entidad proveedora del servicio que le permitirá utilizar dicho servicio.

Esquemáticamente, todo este procedimiento puede verse en la Figura C3.

Figura C3. Esquema del funcionamiento de Jini



## 2.2. PRESENCIA EN EL MERCADO

Jini ha sido desarrollado por *Sun Microsystems*, aprovechando la experiencia y muchos de los conceptos en los que está inspirado el lenguaje Java y, sobre todo, en la filosofía de la Máquina Virtual Java (JVM). Por lo tanto, el Jini puede funcionar sobre potentes estaciones de trabajo, en PC, en pequeños dispositivos (PDA, cámaras de fotos, móviles, reproductores mp3) o en electrodomésticos de línea marrón o blanca (HiFi, TV, Vídeos, *set-top boxes*, frigoríficos, lavadoras, etc.).

Desde su lanzamiento y presentación en el año 1999, la tecnología Jini quizá no esté teniendo el éxito que se esperaba de ella. De hecho, la propia *Sun* así lo ha reconocido. Algunos fabricantes de dispositivos achacan este relativo fracaso a la actitud que mantiene *Sun* respecto a los derechos sobre el Java y su máquina virtual. Aunque cualquier fabricante puede usar el Java en infinidad de aplicaciones de sobremesa o embarcadas, realmente sólo *Sun* o alguna empresa autorizada puede desarrollar la JVM.

Por otro lado, *Microsoft* está contraatacando con el *Universal Plug&Play* (UPnP) el cual se puede montar sobre sistemas operativos usados de forma masiva como Windows, Pocket PC y otros. No hay que olvidar que los usuarios demandan aplicaciones, independientemente de la tecnología que las implemente y, hoy en día la mayoría de las aplicaciones corren sobre sistemas operativos como el Windows 98, Me, 2000, XP, etc ■

### 3. HAVI (HOME AUDIO/VIDEO INTEROPERATIVITY)

HAVI ha sido desarrollado para cubrir las demandas de intercambio de información entre los equipos de audio y vídeo digitales de las viviendas actuales. Es independiente del *firmware* usado en cada uno de los equipos, de hecho, HAVI tiene su propio sistema operativo (independiente del HW y de la función del equipo) que ha sido especialmente diseñado para el intercambio rápido y eficaz de grandes paquetes de datos de audio y vídeo (*streaming*).

Cuando estemos adquiriendo un equipo con el logo HAVI de alguno de los fabricantes que han adoptado el estándar, tendremos asegurado que:

La interoperabilidad será total, cualquier otro dispositivo HAVI podrá gobernar al nuevo y viceversa.

Compatibilidad entre dispositivos de fabricantes diferentes está asegurada.

*Plug&Play* inmediato. Una vez conectado el bus IEEE 1394 al nuevo dispositivo, este se anunciará al resto de equipos HAVI instalados en la vivienda y ofrecerá sus funciones y servicios a los demás. No será necesario estudiarse ningún manual de configuración o de instalación en red del nuevo equipo.

Podremos descargar de Internet las nuevas versiones de SW y controladores que actualizarán las prestaciones del equipo adecuándolo así a las necesidades de cada usuario o a su entorno de equipos HAVI que tenga instalados en su vivienda.

Para todo ello, HAVI ha escogido al estándar IEEE 1394 (también conocido por “*i.Link*” o “*FireWire*”) como soporte físico de los paquetes de datos. Este estándar, que alcanza velocidades de hasta 400 Mbps (o más en las nuevas versiones de IEEE 1394), es capaz de distribuir al mismo tiempo diversos paquetes de datos de audio y vídeo entre diferentes equipos de una vivienda, además de todos los paquetes de control necesarios para la correcta distribución y gestión de todos los servicios.

#### 3.1. DESCRIPCIÓN TÉCNICA

HAVI representa una arquitectura de *software* distribuida, donde no hay un dispositivo maestro que controle el

flujo de información entre los distintos elementos que componen la red. Cada dispositivo de una red HAVI está diseñado para controlar (y ser controlado) a (y por) cualquier otro dispositivo de la red, esté donde esté.

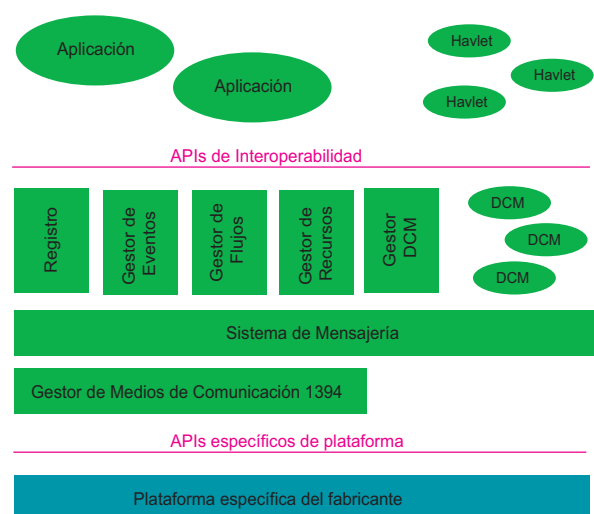
Con el fin de gestionar los comandos y los diferentes flujos de audio y vídeo digital, HAVI utiliza la capa física definida por el estándar IEEE 1394, que proporciona un ancho de banda de 400 Mbps y es capaz de establecer comunicaciones asíncronas que hacen posible la gestión múltiple y simultánea de varios flujos AV en tiempo real.

Los elementos *software* que componen esta arquitectura distribuida son los que soportan los distintos servicios definidos, entre ellos: la gestión de red, la abstracción de dispositivos, la comunicación entre dispositivos y la gestión del interfaz de usuario de los dispositivos.

De manera colectiva, estos elementos *software* muestran sus interfaces de aplicación API interoperables como un conjunto de servicios que pueden ser utilizados y gestionados por todos los dispositivos de la red, de manera que las aplicaciones (que son los elementos *software* en sí mismos) puedan acceder a esos API de manera transparente a lo largo de toda la red.

En la Figura C4 se muestra la arquitectura de capas soportada por HAVI.

Figura C4. Arquitectura en capas y elementos *software* de un dispositivo HAVI



Según esta estructura, vemos cómo los elementos *software* de un dispositivo HAVI forman una capa intermedia entre los API específicos de la plataforma y los API interoperables definidos para soportar las aplicaciones. En esta capa intermedia podemos distinguir los diferentes elementos *software* que comprenden la arquitectura HAVI:

- Gestor de Medios de Comunicación 1394, que controla el flujo síncrono o asíncrono de la información sobre una red IEEE-1394.
- Sistema de mensajería, que controla el flujo de mensajes entre dispositivos.
- Registro, que sirve como directorio de servicios permitiendo a cada elemento *software* localizar al resto de elementos *software* y detectar sus capacidades y propiedades.
- Gestor de Eventos, que controla el envío y recepción de eventos que modifican el estado de los diferentes elementos *software*.
- Gestor de Flujos, quien se encarga de la gestión de las transferencias en tiempo real de los flujos AV digitales.
- Gestor de recursos, que facilita la compartición de los recursos y programa las acciones sobre los mismos.
- DCM (*Device Control Module*). Son elementos *software* asociados a cada uno de los dispositivos HAVI instalados en la red y contienen los API de ese dispositivo y sus capacidades. Los DCM son dinámicos por naturaleza y se instalan o desinstalan cada vez que se inserta o retira un dispositivo en la red.
- FCM (*Function Component Module*). Están contenidos dentro de los DCM y definen cada una de las funciones que realiza el dispositivo. HAVI define los FCM y sus correspondientes API para cada una de las funciones posibles de un dispositivo (sintonización, grabación en cinta, grabación en disco, *display*, filmación, ...).
- Gestor de DCM, quien se encarga de la instalación o desinstalación de los DCM.
- Aplicaciones. Las aplicaciones necesitan hacerse conocer en la red HAVI como si fueran un elemento

*software* para que se permita la comunicación con otros elementos *software* tales como el Registro, los DCM u otras aplicaciones.

Además de estos, HAVI incluye unos elementos *software* especiales que sirven para crear interfaces de usuario sobre dispositivos remotos. Esta característica es una de las claves del HAVI y permite a un usuario acceder a las características de un dispositivo específico de un fabricante, por ejemplo, o interactuar con nuevos DCM que se desarrollen más tarde e instalarlos en un sistema HAVI existente. Estos elementos son los denominados *Havlets*, que son aplicaciones Java que pueden extraerse de los DCM o de una Aplicación, bajo petición de un dispositivo de presentación (*display*) y que, al ser ejecutadas, presentan el interfaz de usuario determinado por el DCM correspondiente mediante un API de interfaz de usuario gráfico (GUI). Para ello, utiliza un protocolo especificado por HAVI y que se denomina DDI (*Data Driven Interaction*).

Todos estos elementos se comunican mediante un mecanismo de envío de mensajes y aunque este mecanismo puede ser implementado de manera distinta por distintos fabricantes, el formato de los mensajes está perfectamente definido por HAVI, garantizando la interoperabilidad.

### 3.2. PRESENCIA EN EL MERCADO

HAVI es una iniciativa de los fabricantes más importantes de equipos de entretenimiento (*Grundig, Hitachi, Panasonic, Philips, Sharp, Sony, Thomson y Toshiba*) para crear un estándar que permita compartir recursos y servicios entre los televisores, los equipos HiFi, los vídeos, etc.. Desde este punto de vista HAVI es una especificación *software* que permite la interoperabilidad total entre dispositivos de estos fabricantes.

Además el estándar HAVI ha definido (y aún está haciéndolo con otras tecnologías) un conjunto de normativas para interactuar con otras iniciativas de normalización en el mundo del “*home networking*”:

- Internet: HAVI define un FCM con APIs que permiten a las aplicaciones HAVI formar parte de los protocolos internet más comúnmente utilizados como HTTP, FTP, POP3,...
- Jini: Los principales miembros de HAVI, junto con *Sun Mycrostem*, han creado un “*bridge*” o puente

que permite a las aplicaciones HAVI interactuar con redes basadas en el estándar Jini.

- UPnP y HomeAPI: De la misma forma que con Jini, se han creado interfaces de programación (APIs) y un “*bridge*” HAVI-HAPI que permite la interoperabilidad entre una red HAVI y una red UPnP basada en Windows.
- *Bluetooth* y HomeRF: Existen algunas iniciativas de interactuar con estas tecnologías en el sentido de poder controlar los *displays* de los dispositivos Bluetooth y HomeRF como si se trataran de dispositivos HAVI.

En líneas generales la organización HAVI está en continuo contacto con otras iniciativas (OSGI, MHP,...) para establecer los procesos de interacción entre dispositivos HAVI y cualquiera de las tecnologías y estándares del mundo del “*home networking*”.

La principal ventaja de HAVI, es que los usuarios pueden usar, por ejemplo, la pantalla del televisor para gobernar el equipo HiFi de música, la vídeo cámara, la videoconsola a la vez que la TV, o pueden escuchar la música del reproductor de CDs del salón en el equipo midi de la habitación, o usar un ordenador, situado en otra habitación, como reproductor de películas DVD mientras cenamos en la cocina o en el salón. Por otro lado, todos estos equipos podrán bajar automáticamente el volumen cuando suene el teléfono o llamen a la puerta. El sistema de alarma de la vivienda podrá usar la TV como pantalla y el vídeo como sistema de almacenamiento.

A simple vista HAVI parece que podría representar el despegue de las tecnologías de webcasting y la televisión interactiva a gran escala ya que permite la distribución de audio y vídeo por toda la casa de manera completamente intuitiva. Sin embargo, HAVI cuenta con dos limitaciones importantes:

- Escaso número aún de fabricantes que lo soportan.
- Limitación a la capa física representada por el IEEE 1934, e interoperabilidad incierta con tecnologías wireless, actualmente más solicitadas ■

## 4. OSGI (OPEN SERVICES GATEWAY INITIATIVE)

La asociación *Open Services Gateway Initiative* (OSGI) nació en marzo de 1999 con el objetivo de crear una especificación *software* abierta, y libre de royalties, que permita diseñar y construir plataformas compatibles que sean capaces de proporcionar múltiples servicios en el mercado residencial y automovilístico. Para ello, aprovecha las múltiples tecnologías que han ido apareciendo en el campo de los métodos de acceso tanto en el ámbito de las redes de datos y control de las viviendas como en el mercado de la telefonía móvil y en el de la automoción.

Con todo ello, el OSGI pretende ofrecer una arquitectura completa y extremo-a-extremo, que cubra todas las necesidades del proveedor de servicios, del cliente y de cualquier dispositivo instalado en las viviendas.

A esta plataforma le han puesto el nombre de “Pasarela de Servicios” en su modalidad genérica, pero en el ámbito de las viviendas, SOHOs y pequeños negocios, se la conoce con el nombre de Pasarela Residencial.

Las áreas en que se vuelcan todos los esfuerzos del OSGI son:

### 1. Servicios

Se pretende crear una plataforma que sea capaz de procesar y tratar de forma correcta toda la información necesaria para proporcionar servicios de comunicaciones, de entretenimiento, de telecontrol o teledomótica, y de seguridad. Por lo tanto, la especificación OSGI debe tener los interfaces adecuados para soportar todos estos servicios sin incompatibilidades además de permitir gestionarlos de forma adecuada.

### 2. Métodos de acceso

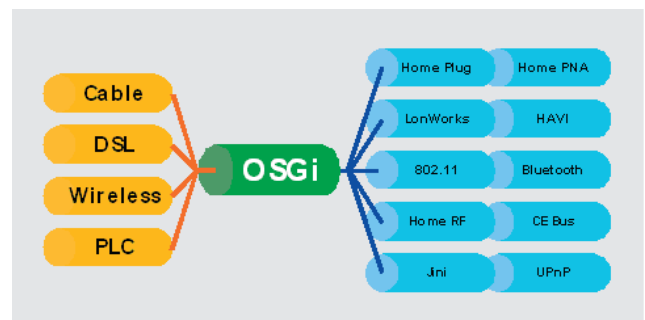
La idea es que la pasarela OSGI sea capaz de acceder al mundo exterior (redes de datos tipo Internet) usando cualquiera de las tecnologías disponibles actualmente. Si bien es cierto que en el año 1999 se contemplaba el uso de métodos de acceso de banda estrecha como los modems RTC, RDSI, GSM, entre otros, ahora la tendencia es volcar todos los esfuerzos en tecnologías de acceso de banda ancha con conexión permanente Internet (Always-On). Destacan el ADSL, el modem de Cable, el PLC o tecnologías inalámbricas WiFi, 3G, LMDS. Esta tendencia hacia la banda ancha, es debido, además del aumento

de la implantación, a la mejora sustancial en la “Calidad del Servicio”, que será un requisito necesario para el éxito del mercado de los e-servicios.

### 3. Redes de datos y control de las viviendas

Teniendo en cuenta la variedad de hogares y edificios en donde este tipo de pasarelas debe ser instaladas, esta iniciativa no escoge una única tecnología de conexión en red para los múltiples electrodomésticos o dispositivos de las viviendas. Su objetivo es definir un interface común para todas ellas, dejando la responsabilidad a los fabricantes de construir los controladores adecuados para cada una de ellas. Teniendo en cuenta esto, la pasarelas OSGI podrán usar tecnologías conexión inalámbricas (IrDa, HomeRF, IEEE 802.11x, *Bluetooth*), sobre cables telefónicos (HomePNA), sobre la red de baja tensión (*HomePlug*, *Lonworks*, EIB/KNX, etc), sobre conexiones como Ethernet, USB, etc, y protocolos como el HAVI, el UPnP, el Jini, etc. Por lo tanto, la especificación OSGI será la “pasarela” que transforme los paquetes de información procedentes del mundo exterior a un paquete de datos de cualquiera de estas tecnologías y viceversa.

Figura C5. OSGI como estándar de pasarelas residenciales



### 4.1. CARACTERÍSTICAS PRINCIPALES

La especificación del OSGI define un núcleo de APIs (*Application Program Interfaces*) que son el principal soporte de los e-services.

Las características principales de la especificación son las siguientes:

**Estandarizada.** Para que los fabricantes de equipos y los proveedores de servicios tengan una plataforma común sobre la que ofrecer sus servicios e impedir que un único fabricante monopolice el mercado.

**Independiente del hardware.** La tecnología tiene que ser independiente de la plataforma *hardware*, de forma que pueda funcionar con soluciones múltiples en el ámbito de los procesadores, las comunicaciones, los electrodomésticos, las soluciones domóticas, etc.

**Abierta.** No define ninguna arquitectura de red domótica ni obliga al uso de una tecnología concreta, ni ningún protocolo. Cualquier empresa puede apostar por introducir sus propia tecnología al producto final guardándose que sea compatible con las APIs predefinidas.

**Segura.** Se define una arquitectura *software* que proporciona una alta seguridad e integridad para que los proveedores puedan ofrecer múltiples servicios sobre la misma plataforma sin interferirse unos con otros.

**Fiable.** La pasarela debe funcionar 24 horas al día, sin caídas del sistema por descuidos o provocadas malintencionadamente.

**Escalable.** La administración y operación del parque de pasarelas, que podría llegar a millones de abonados, debe ser flexible, personalizable y escalable acorde a las nuevas necesidades del proveedor del sistema.

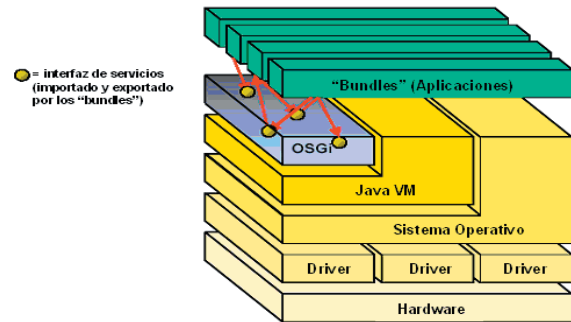
## 4.2. DESCRIPCIÓN BÁSICA DEL FUNCIONAMIENTO

La arquitectura de OSGI tiene dos elementos fundamentales, por un parte una plataforma de ejecución de aplicaciones basada en Java y por otra, una serie de paquetes (“*bundles*”) que proporcionan una determinada funcionalidad a otros paquetes, o directamente al usuario final, y que se ejecutan sobre la plataforma.

Estos elementos residen siempre en un elemento central que se denomina Plataforma de Servicio OSGI situada en la red local y conectada al proveedor de servicios a través de una pasarela en la red del operador. Este elemento además será el encargado de permitir la interacción entre dispositivos o redes de dispositivos que empleen distintas tecnologías para comunicarse.

En OSGI un servicio está proporcionado por un bundle que se ejecuta en la plataforma OSGI, cuya estructura funcional se muestra en la Figura C6.

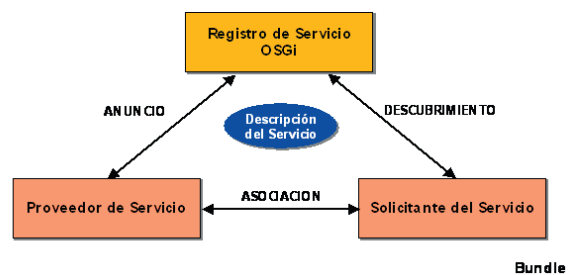
Figura C6. Componentes de una plataforma de servicios OSGI



En la plataforma existe el Registro de Servicio OSGI que actúa como un servicio de directorios en el que los bundles se registran y a través del cual pueden localizar a otros bundles para componer nuevos servicios.

En la especificación de OSGI se han definido una serie de APIs básicas para el desarrollo de servicios, como el de logging, servidor HTTP y el que se denomina Especificación de Acceso al Dispositivo (DAS-Device Access Specification), que permite el descubrimiento y anuncio dinámico de dispositivos y de los servicios ofrecidos por éstos (Figura C7).

Figura C7. Mecanismo de anuncio y descubrimiento en OSGI



Con el mecanismo DAS se definen los Bundles de Red que son los encargados de descubrir nuevos dispositivos y servicios empleando el protocolo de descubrimiento correspondiente en esa red, una vez obtenido esta información, deben obtener del proveedor de servicios el Bundle de Dispositivo correspondiente al dispositivo concreto,

que se instalará en la plataforma y se registrará en Registro de Servicio OSGI, conteniendo la descripción del servicio, y de esta forma podrá ser descubierto por otros bundles o por otros solicitantes del servicio, creando la asociación correspondiente con el proveedor del mismo para que puedan interactuar.

Mediante este mecanismo, OSGI permite la integración de cualquier protocolo de descubrimiento y anuncio de servicios dentro de su plataforma, como los definidos por UPnP o Jini.

### 4.3. PRESENCIA EN EL MERCADO

El OSGI es un grupo de trabajo que surgió en Marzo de 1999, cuyo principal impulsor es Sun Microsystems. El resto de los miembros fundadores son: *Alcatel, Cable and Wireless, Electricité de France, Enron Communications, Ericsson, IBM, Liberate Technologies, Lucent Technologies, Motorola, Nortel Networks, Oracle, Philips Electronics, Sybase* y *Toshiba*. Actualmente ya tiene socios españoles como Unión Fenosa y Telefónica I+D.

Desde esa fecha, la iniciativa OSGI ha desarrollado distintas versiones del estándar, en las que se van añadiendo más funcionalidades y protocolos de interoperación con otras tecnologías y estándares. La última de estas versiones disponibles, la Release 3, incorpora el soporte para plataformas de servicio y aplicaciones móviles, así como una nueva sección de arquitecturas de referencia que permiten, de manera más sencilla, el desarrollo de servicios sobre plataformas OSGI basados en especificaciones complementarias como Jini o UPnP.

La iniciativa OSGI sólo pretende definir un arquitectura *software* que sea independiente del hardware usado (microprocesador, memoria, periféricos, modems, etc). Ahí radica una de sus principales ventajas ya que permitirá a cualquier fabricante decidir cómo y donde embarca este *software*.

Por lo tanto, la pasarela OSGI aparecerá en multitud de dispositivos “inteligentes”, ya sea de forma completa o de forma parcial, dependiendo del fabricante o del proveedor de servicios en particular. Podremos encontrarla dentro de *set-top-boxes* de televisión por cable, por satélite o terrestre digital, dentro de modems de Cable, routers, centralitas de alarma, contadores electrónicos de electricidad, en la puerta de un frigorífico, e incluso, como un simple *software* para ordenadores de sobremesa.

Aunque inicialmente la intención del OSGI era orientarse al mercado residencial ahora también se está trabajando para embarcar la especificación dentro de los automóviles (que también es una de las aportaciones más relevantes de la Release 3), ya que cada vez más, estos cuentan con multitud de sistemas de ayuda a la navegación o para el entretenimiento que serán ideales para ofrecer nuevos servicios bajo demanda. Incluso, una pequeña parte de esta especificación podrá algún día ser embarcada dentro de las agendas personales o PDAs y en los teléfonos móviles.

Pero la versión que parece que saldrá antes al mercado, serán las que ya están preparando algunos fabricantes como *Ericsson, Domsys, 2wire, Cisco, Sun, IBM*, entre otros, que será una mezcla de router y videconsola y dispondrá de varios interfaces para redes internas de datos y control, además de una conexión a la roseta telefónica (ADSL) o a la entrada del modem de Cable. Estarán diseñadas para actuar como servidor multiservicios ■

## 5. MHP (MULTIMEDIA HOME PLATFORM)

MHP es una iniciativa de estandarización que trata de especificar las aplicaciones interactivas de vídeo y TV digital, permitiendo una elevada portabilidad e interoperabilidad entre descodificadores y plataformas. Sus aportaciones se basan en la experiencia tecnológica acumulada en el campo de la Televisión Digital durante los últimos años.

En 1997 nació DVB-MHP, una actividad lanzada por el Proyecto DVB (*Digital Video Broadcasting*) con el objetivo de estandarizar los elementos que conforman una plataforma en el hogar, clave para el éxito de las aplicaciones multimedia. Y aunque inicialmente esta plataforma fue pensada para todo tipo de aplicaciones multimedia, rápidamente se produjo una evolución lógica en el proceso de estandarización que provocó que el principal foco de atención se centrara en las aplicaciones interactivas para los sistemas de Televisión Digital.

En febrero de 2000, en el vigésimo octavo encuentro del *DVB-Steering Board* se presenta la primera especificación del estándar, MHP 1.0. Este grupo asume la responsabilidad de tomar las decisiones finales sobre todos los problemas referentes a la norma, y participa en las discusiones con los poseedores de las tecnologías clave relacionadas con la plataforma.

Poco más tarde, en el mes de junio del mismo año, esta especificación es adoptada por el ETSI (*European Telecommunications Standards Institute*) con la referencia TS 101 812 (hoy día formalizada ya en la especificación ETSI ES 201 812 V1.1.1). Sin embargo, el esfuerzo de estandarización no se detuvo, y continuó evolucionando a través de los diversos “perfiles” que DVB-MHP define, y como resultado, en noviembre de 2001 la ETSI publica la especificación DVB-MHP, con la referencia TS 102 812, que corresponde a lo que se ha denominado MHP 1.1 y que cubre todos las áreas indicadas en el estándar.

A partir de ese momento el desarrollo del estándar MHP se extiende hacia la definición de los procesos por los cuales se puede obtener la certificación MHP de productos concretos con la creación de un conjunto de tests contenidos en la especificación “MHP Test Suite”, así como de maximizar la interoperabilidad entre el MHP y otras organizaciones normativas, como por ejemplo la liderada por CableLabs, quien en Enero de 2002 anuncia la adopción del MHP para OCAP. El resultado final de este trabajo se ve reflejado en el GEM (*Globally Execu-*

*tible MHP*), y su publicación por ETSI con la referencia TS 102 819, que no es una especificación que pueda ser directamente implementada aisladamente sino que muestra un entorno de trabajo a utilizar por las distintas organizaciones que deseen definir sus especificaciones basadas en MHP.

### 5.1. DESCRIPCIÓN TÉCNICA

DVB-MHP ha especificado una plataforma estándar basándose en el conocimiento acumulado de experiencias anteriores y tratando de proveer mecanismos que faciliten su adopción en el mercado de la forma menos traumática posible.

Para ello, sus principios de funcionamiento se basan en la definición de unos perfiles que marcan la evolución de la plataforma, junto una arquitectura y unos procesos flexibles pensados para facilitar la portabilidad e interoperabilidad de aplicaciones, que están sometidas a un ciclo de vida muy definido.

#### 5.1.1. Perfiles

Durante la especificación del MHP se ha tenido en cuenta su posible evolución, por lo que se han diseñado un conjunto de perfiles que definen sus características y que vienen marcados por distintas áreas de actuación o “*profiles*” que van incorporando, de manera incremental, mayores y mejores prestaciones.

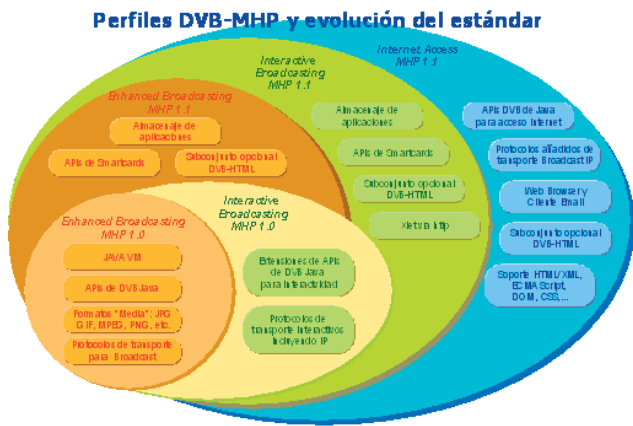
Un “*profile*” hace referencia a un área de aplicación, o, como consecuencia, a un conjunto de prestaciones de los *set-top-boxes*. Se distinguen tres “*profiles*” o áreas de aplicación:

- En primer lugar, se define el área denominada “*Enhanced Broadcasting*,” que combina la transmisión digital de los servicios de vídeo y audio del operador con las aplicaciones que pueden ser descargadas para ser ejecutadas en el descodificador ofreciendo interactividad local; es decir, estas plataformas no poseen canal de retorno y la única interactividad que ofrecen se desarrolla en el propio descodificador del usuario. (Perfil definido en MHP 1.0)
- En segundo lugar, se sitúa el área correspondiente a “*Interactive Broadcasting*”, en la que se pueden

proporcionar servicios interactivos que pueden o no estar asociados a los servicios de vídeo y audio ofrecidos por el operador, y que permiten una interactividad completa. En este caso, la plataforma está dotada de un canal de retorno, con objeto de realizar la comunicación entre los descodificadores y la cabecera. (Perfil definido en MHP 1.0)

- Finalmente se define el área llamada “Internet Access”, cuyo objetivo es permitir proveer a los usuarios de servicios Internet, y que como nota destacada contiene un elemento HTML opcional denominado DVB-HTML. (Perfil definido en MHP 1.1)

Figura C8. Evolución de las versiones de MHP y sus perfiles



que incluso el operador pueda controlarlo desde la cabecera. Para llevar a cabo esta tarea, la capa de *software* de sistema incluye un elemento más: el *Application Manager*. Cuando un operador solicita un cambio de estado en una aplicación, o simplemente el sistema necesita detener una aplicación, la petición llega al *Application Manager*, que se encarga de ejecutar la orden recibida.

Finalmente, en la parte superior de la arquitectura, aparece la Capa de Aplicaciones, que es lugar donde se sitúan las diversas aplicaciones descargadas y que utilizan los servicios ofrecidos por las capas inferiores.

Realmente, a través de esta arquitectura en capas se está estableciendo una relación jerárquica en la que el usuario, a través de la interfaz que la aplicación le ofrece, controla su ejecución, y la aplicación, para poder llevar a cabo su cometido, utiliza el API disponible gracias a la capa de *software* de sistema, que finalmente acaba utilizando los recursos disponibles en el descodificador.

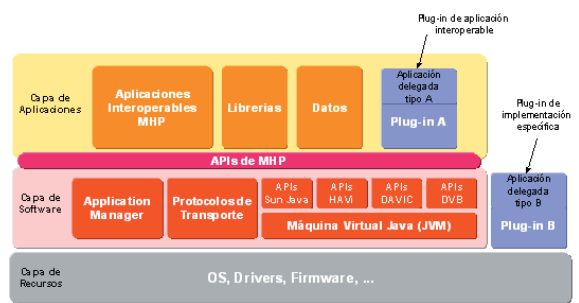
Sin embargo, el esfuerzo de estandarización no se queda exclusivamente en el hecho de especificar una API para los desarrolladores de aplicaciones interactivas, sino que llega a definir el comportamiento de dichas aplicaciones. Mediante la definición de su ciclo de vida, MHP permite controlar los diversos estados en los que las aplicaciones se pueden encontrar durante su ejecución, evitando en todo momento la aparición de situaciones incontroladas y confiriendo a la ejecución una gran robustez y fiabilidad.

### 5.1.2. Arquitectura

Con el fin de lograr los objetivos de interoperabilidad y de compatibilidad con versiones (o perfiles) anteriores, la plataforma se organiza siguiendo una arquitectura en capas (Figura C9), siendo la inferior la Capa de Recursos del descodificador. Sobre ésta se sitúa la Capa de *Software* de Sistema, que aislará a las aplicaciones de los recursos existentes en el descodificador. Es en esta capa donde se implementa la JVM (*Java Virtual Machine*), que ofrece las diversas bibliotecas Java que finalmente conformarán el API que los proveedores de aplicaciones interactivas podrán utilizar en sus desarrollos. Y es precisamente este API el objeto de la mayor parte del esfuerzo de especificación llevado a cabo por DVB-MHP.

Sin embargo, MHP define también el comportamiento que las aplicaciones deben presentar en su ejecución, es decir, el ciclo de vida de las aplicaciones, permitiendo

Figura C9. Arquitectura de capas de la especificación MHP



### 5.1.3. Modelos de aplicaciones

En MHP se definen dos modelos de aplicaciones que están claramente diferenciados. Por un lado, se presentan las aplicaciones DVB-J, basadas en tecnologías Java, y,

por otro, las aplicaciones DVB-HTML, que se basan en el uso de HTML y JavaScript. Aunque entre ambos modelos existen algunas diferencias, se presentan algunas características comunes a todas ellas.

En primer lugar, como ya se ha mencionado, las aplicaciones presentan un ciclo de vida muy definido, evitando el paso por estados incontrolados que podrían llevar a la plataforma a inestabilizarse. Aún así, para mayor seguridad, se proveen mecanismos que permiten que el operador pueda controlar el estado de las aplicaciones mediante la señalización desde la cabecera. En segundo lugar, MHP permite la concurrencia, es decir, un mismo terminal puede estar ejecutando varias aplicaciones al mismo tiempo. Esto implica un control de acceso a los recursos, que se realiza a través del arbitrio del *Application Manager*.

Finalmente, las aplicaciones están asociadas a un servicio, es decir, a un canal. En el momento en el que el usuario cambia de canal, la plataforma consulta qué aplicaciones debe ejecutar y procede en consecuencia. Por ello, MHP especifica un mecanismo para la señalización de aplicaciones permitiendo llevar a cabo esta operativa.

De esta manera el desarrollador de aplicaciones solo tiene que preocuparse de utilizar las API disponibles. En el caso de aplicaciones DVB-J, estas son interfaces Java, recogidas de múltiples fuentes, como DAVIC, HAVI, JavaTV, y algunos de nuevo desarrollo como el paquete org.dvb.si. Para las aplicaciones DVB-HTML, se utiliza tecnología HTML y ECMAScript, haciendo visible la API Java a este último para poder llevar a cabo el acceso a la plataforma.

### 5.1.3. Señalización de aplicaciones MHP

En una plataforma MHP las aplicaciones están asociadas a un servicio. Se debe suministrar, por tanto, un mecanismo que permita llevar a cabo esta asociación, y que a su vez el decodificador pueda interpretar.

Este mecanismo es la Señalización de Aplicaciones, que tiene como objetivos la identificación y localización de las aplicaciones asociadas a un servicio, el control de su ciclo de vida desde la cabecera y la identificación de las fuentes de datos requeridas por las aplicaciones de un servicio.

El flujo de transporte de un sistema de TV Digital contiene unas tablas denominadas PSI (*Programm Specific*

*Information*) que permiten realizar la decodificación del vídeo y audio emitidos. Una de estas tablas es la PAT (*Programm Association Table*), que contiene los identificadores de otras tablas descriptoras de cada uno de los programas que son transportados: las tablas PMT (*Programm Map Table*). Éstas realizan la descripción de cada uno de los programas que son transportados indicando sus componentes de audio, vídeo y datos, permitiendo de esta forma realizar su decodificación.

Para poder identificar las aplicaciones que están asociadas a un servicio, se introduce un nuevo descriptor en la PMT, denominado "*application\_signalling\_descriptor*", que contiene el identificador (PID) que permite localizar una nueva tabla: la AIT (*Application Information Table*). Por tanto, existe una AIT para cada programa o servicio que es emitido en un "*transport stream*". Esta tabla contiene la lista de aplicaciones que están asociadas al servicio al que hacen referencia. Para poder realizar el control del ciclo de vida para cada una de las aplicaciones incluidas en la tabla se definen unos códigos de control ("*application\_control\_code*"), cuyo valor hace que la aplicación realice una transición al estado indicado.

Finalmente, y con el objetivo de identificar las fuentes de datos que las aplicaciones necesitan, se incluye el "*transport\_protocol\_descriptor*", que hace referencia al protocolo de transporte utilizado, y el "*dvb\_html\_application\_boundary*", que permite definir los límites de una aplicación DVB-HTML. En el caso de las aplicaciones DVB-J estos límites quedan definidos a través de la señalización de extensiones para la variable de entorno CLASSPATH.

### 5.1.4. Funcionalidades añadidas: los *plug-ins*

La arquitectura de las plataformas MHP se completa con la capacidad de admitir *plug-ins* (Figura C9), que aportan una gran flexibilidad a la misma. Un *plug-in* se define como un conjunto de funcionalidades que pueden ser añadidas a la plataforma, de tal forma que sea capaz de interpretar aplicaciones y formatos de datos que no han sido definidos en la especificación.

Mediante el concepto de *plug-in* se resuelven dos problemas simultáneamente. Por un lado, se consigue que un amplio espectro de aplicaciones que han sido desarrolladas hasta la fecha sobre otras plataformas puedan llegar a funcionar en una plataforma MHP, facilitando en cierta medida la adopción del estándar a través de una transición suave hacia el mismo. Por otro lado, si se for-

zase la situación obligando a que todas las plataformas MHP funcionen de la misma forma y con las mismas características, se estarían eliminando elementos diferenciadores que al fin y al cabo son los que permiten competir a los proveedores de plataforma. Utilizando los *plug-ins*, éstos, pueden incluir en su plataforma ciertas funcionalidades que otros proveedores no ofrecen, presentando de esta forma un elemento diferenciador con el cual competir.

La selección de *plug-ins* se debe dejar a elección de los usuarios, para que estos sean libres a la hora de escoger la fuente de servicios que utilizan. Esto se puede conseguir a través de diversos mecanismos:

El usuario podría adquirir un equipo MHP en el que el *plug-in* venga suministrado de fábrica, de tal forma que su plataforma dispone de unas características particulares.

Otra opción sería posibilitar la descarga de un determinado *plug-in* a través de una petición del usuario cuando éste necesita ejecutar una aplicación o interpretar un formato de datos que la plataforma no soporta.

Y finalmente este proceso podría ser automatizado, siempre en función de los recursos disponibles en la plataforma.

Sin embargo, quizás la mejor solución consista en una combinación de las tres posibilidades expuestas.

Para poder llevar a cabo su función, estos *plug-ins* deben encajar de alguna forma en la arquitectura de la plataforma, y efectivamente lo hacen situándose en alguna de las capas de *software* que se definen, clasificándose de este modo. Por un lado, se tienen los denominados *Plug-ins Interoperables*, que son desarrollados utilizando la API Java de la plataforma y que, por tanto, se sitúan en la capa de aplicaciones. Por otro, se definen los *Plug-ins de Implementación Específica*, que se realizan en código nativo, y que pueden implementar una API Java no especificado, pudiendo utilizar las aplicaciones que se desarrollan sobre éstos.

Dadas, las fuertes restricciones de seguridad a las que está sometida la plataforma, y que están fundamentadas básicamente en el concepto de “*sand-box*”, los *plug-ins* son los responsables de velar por la seguridad de las aplicaciones que ejecutan, autenticándose de forma apropiada cuando se necesita acceder a recursos que necesitan de la concesión de permisos especiales

En resumen se puede concluir que, DVB-MHP ha especificado una plataforma estándar basándose en el conocimiento acumulado de experiencias anteriores y tratando de proveer mecanismos que faciliten su adopción en el mercado de la forma menos traumática posible. Para ello, sus principios de funcionamiento se basan en la definición de unos perfiles que marcan la evolución de la plataforma, junto una arquitectura pensada para facilitar la portabilidad e interoperabilidad de aplicaciones, que están sometidas a un ciclo de vida muy definido.

## 5.2. CONSIDERACIONES DE MERCADO

### 5.2.1. Justificación

La falta de normalización ha provocado que cada proveedor desarrolle su plataforma de servicios interactivos de forma independiente, cumpliendo con las normas existentes, pero implementando de un modo propietario las API que luego utilizarán los proveedores de servicios interactivos (y obviando por supuesto los problemas derivados del uso de distintos medios de transmisión).

Ante esta situación el trabajo realizado por DVB-MHP se puede interpretar, desde un punto de vista global, como un intento de estandarización de las API que van a utilizar los proveedores de aplicaciones interactivas en sus desarrollos. La clave del éxito reside en conseguir un aislamiento de estas aplicaciones respecto de los recursos de los descodificadores. Si, además, la API utilizada para desarrollar las aplicaciones se adopta por todos los implicados, estas aplicaciones serían compatibles con cualquier plataforma del mercado. El resultado de este esfuerzo de estandarización sería la obtención de una plataforma digital, compatible DVB, que presenta una API de programación unificada, con la que las aplicaciones serían intercambiables y que sería capaz de ser ejecutada en la plataforma de cualquier operador.

Este hecho tiene unas implicaciones enormes. En primer lugar, además de la disminución de los costes de desarrollo de los proveedores de servicios interactivos, se consigue ofrecer una gran movilidad a los usuarios finales, que libremente podrían acudir a un punto de venta a seleccionar un descodificador digital, en función de sus preferencias. Con este descodificador único, los usuarios podrían abonarse a cualquiera de las ofertas existentes en el mercado, y en el caso de querer estar abonado a otro operador no necesitarían disponer de una pila de descodificadores en su hogar. En definitiva, el mercado adquiere una estructura horizontal.

Aunque bien es cierto, que esta nueva estructura se deberá enfrentar a los diferentes medios de transmisión existentes, a los distintos sistemas de acceso condicional etc. La situación deberá evolucionar, y quizás acabe en un punto, en el que se disponga de una fuente común de contenidos (de todos los contenidos) que podrá ser recibida en la ciudad a través de cable, en el campo a través del satélite, y cuando uno se va de camping y playa, a través de la televisión terrestre.

Además, como consecuencia a largo plazo, esta nueva estructura horizontal, puede provocar grandes cambios en el flujo de relaciones que se establecen en el modelo de negocio de la TV Digital, en el que el operador es el cliente de los proveedores de los equipos de cabecera e infraestructura, de los proveedores de la plataforma de servicios interactivos y de los proveedores de aplicaciones interactivas, lo que produce una gran concentración de las inversiones y de la toma de decisiones en un mismo punto.

### 5.2.2. Mercado horizontal

La estructura horizontal que adoptaría el mercado con MHP, y la interoperabilidad conferida a los servicios interactivos, provocaría que éstos pudiesen ser desarrollados independientemente del operador que los va a emitir, favoreciendo que organizaciones ajenas desearan disponer de un servicio interactivo en un canal de televisión. Una compañía de comida rápida a domicilio podría decidir que desea disponer de una aplicación de venta en la televisión durante la emisión de un programa determinado. En lugar de acudir a un operador, esta compañía, acudiría a una central de medios o a los proveedores de servicios interactivos, que desarrollarían la aplicación según los gustos y preferencias de su cliente. Cuando la aplicación hubiese sido terminada, se acudiría a cualquiera de los operadores que ofrecen servicio, y pactaría un coste por el ancho de banda ocupado por la aplicación. Finalmente la aplicación sería emitida en los horarios pactados.

Este nuevo flujo de relaciones ofrece claras ventajas para el impulso y la penetración en el mercado de la Televisión Digital, que finalmente acabarían beneficiando a todos. En primer lugar, el motor de aplicaciones interactivas, y en definitiva, el impulsor económico de estos desarrollos sería cualquier entidad que deseara poner en emisión una aplicación, sin estar condicionada por las decisiones y circunstancias particulares de un operador. Además, con el nuevo modelo, el espectro de potenciales

clientes de una empresa proveedora de servicios interactivos se ensancha enormemente, y se aleja de los factores de riesgo con los que se mueven ahora los operadores, y que en muchas ocasiones frenan el impulso del desarrollo de aplicaciones.

En segundo lugar, la proliferación de más y mejores aplicaciones interactivas provocaría una rápida evolución de la tecnología, que combinada con una mayor tasa de penetración en el mercado, conseguiría que las inversiones de los operadores en infraestructura fuesen amortizables en un plazo menor, además de animar el mercado de los proveedores de plataforma, acceso condicional o fabricantes de descodificadores.

Por último, la inversión realizada no estaría concentrada exclusivamente en los operadores, sino que se produciría una distribución de la misma entre los diversos negocios que intervienen en el proceso, si bien es verdad, que las inversiones más costosas las seguirían realizando aquéllos. Además, el operador tendría como objetivo la difusión de contenidos, recibiendo unos ingresos, además de por el número de abonados, por el alquiler del ancho de banda utilizado por las aplicaciones. De esta forma, se crea una nueva forma de ingresos por servicios interactivos.

En definitiva, el esfuerzo de estandarización que DVB-MHP, que incluso llega a definir el comportamiento de ejecución de las distintas aplicaciones en los descodificadores, nace como respuesta a una situación actual, tratando de conseguir que la penetración de la TV Digital en el mercado mejore, beneficiando a todos.

También es cierto que la estandarización de una plataforma hace que ésta tenga que ser abierta, y que el control sobre su evolución lo tome el organismo de estandarización, situación que probablemente no favorece a los proveedores de plataforma actuales. Este hecho, unido a la habitual lentitud que conlleva la definición de estándares comparada con la rápida evolución del mercado (que demanda cubrir necesidades ya reales) es lo que ha impedido que hasta ahora no exista todavía una amplia y extendida oferta real de dispositivos y plataformas MHP.

No obstante, las expectativas indican que MHP será un estándar de alto cumplimiento a medio plazo, una vez que las plataformas actuales hayan evolucionado, y de hecho, la gran mayoría de fabricantes ya han anunciado su compatibilidad con las primeras versiones del estándar, o su adopción en las siguientes versiones de los productos actuales ■

---

Figura C10. Mapa de adopción del estándar DVB-MHP

---

