



# AwoX Solution Guide

Software Development Kits and Tools  
for creating DLNA-compatible  
multimedia applications and devices

## CONTENTS

Introduction .....	5
Presenting AwoX .....	5
What is the Digital Living Network Alliance? .....	5
Available AwoX mediaCTRL SDKs.....	7
DLNA use cases and the AwoX offer .....	8
Media use cases.....	9
Storage and printer use cases .....	10
Internet and premium use cases.....	11
Advanced use cases .....	12
Players and servers .....	13
Controllers and renderers .....	14
Upload and download .....	15
DLNA device capabilities: upload, printer, Remote User Interface (RUI).....	16
Conclusion.....	16
mediaCTRL middleware architecture overview .....	17
mediaCTRL Controller/Player SDK.....	17
High Level APIs .....	18
APIs for optional components .....	19
Controller SDK performance indicators.....	20
mediaCTRL Server Plug-in SDK.....	21
How the Server Plug-in works.....	22
High Level APIs .....	22
Media server management API.....	23
Plug-in Framework.....	23
Server Plug-in SDK performance indicators .....	24
mediaCTRL Renderer SDK .....	24
High Level APIs .....	25
Media player sample code .....	25
Renderer SDK performance indicators.....	25
mediaCTRL DTCP SDK .....	26
What is DTCP .....	26
Overview of mediaCTRL DTCP Source and Sink SDKs .....	26
DCTP Source-side architecture and integration .....	27
DCTP Sink-side architecture and integration .....	29
DTCP certification.....	29

Using S-Bus for inter-SDK communication .....	30
Working with AwoX .....	32
What is included in the AwoX mediaCTRL SDKs?.....	32
Target operating system requirements .....	32
Getting started with your project .....	33
Frequently Asked Questions.....	33
Certification and testing.....	35
Compliance + Interoperability = Certification .....	35
Clearly defining your product profile.....	35
Available test resources.....	36
Plug-Fests .....	37
UPnP certification .....	37
Wi-Fi certification .....	37
Working with Independent Certification Vendors.....	38
Appendix - mediaCTRL SDK details .....	39
Parameters.....	39
SSDP configuration.....	39
UPnP configuration.....	39
ControlPoint configuration.....	40
Web server configuration .....	40
Web client configuration.....	40
SDK Contents.....	42
AwoX mediaCTRL SDKs .....	42
UPnP Logger .....	42
Porting requirements.....	43
Device classes and supported operating systems .....	44
Supported platforms .....	45
Stack details .....	46
Streaming protocols.....	47
Media players .....	48
Media server.....	49
Glossary.....	50

© 2011 AwoX. All rights reserved. mediaCTRL is the exclusive property of AwoX. All other brands mentioned are the property of their respective owners.

*Ref: AwoX Solution Guide 1.0 (November 2011)*

## Introduction

Welcome to the AwoX world of connected multimedia integration! This document presents the software tools and components provided in the AwoX mediaCTRL<sup>®</sup> Software Development Kits (SDKs). Here we offer a general introduction to the context and concepts related to the AwoX solutions. The document then progresses towards a more detailed technical description of the offer and available software components that you can integrate into your own product designs as a multimedia integrator.

## Presenting AwoX

AwoX provides advanced software solutions that enable consumer electronics and mobile device companies to embed standard and interoperable multi-media features in their products with minimal development effort. These fully DLNA (Digital Living Network Alliance) compatible solutions are designed to be integrated into a wide variety of hardware platforms. As *modular components*, AwoX solutions are provided to you in the form of its mediaCTRL Software Development Kits (SDKs) comprised of code libraries, application programming interfaces (APIs), documentation, and reusable samples. The specific mediaCTRL SDKs you need for your own projects will depend on the target end-product and desired functionality.



A reference supplier in the DLNA world, AwoX provides customers with ready-to-use functionality that would otherwise require significant internal training, development, and testing efforts. The goal of mediaCTRL SDK solutions is to provide customers with pre-certified DLNA compatible software that abstracts away the myriad minute details of audio and video playback, file browsing, client and server identification, and much more, enabling you to concentrate on more value-added aspects of multimedia product development. This approach improves application reliability, expands feature sets, and speeds up time-to-market in a cost-effective manner for new products. The functionality provided by mediaCTRL SDKs respects DLNA Interoperability Guidelines while providing a solid foundation that is flexible enough for you to focus on product differentiation.

## What is the Digital Living Network Alliance?

The Digital Living Network Alliance has published a set of functional guidelines to enable many different types of multimedia products and devices to interact with each other transparently in the “connected home.” These guidelines are driven by examining the vast variety of potential (and continually evolving) use cases, and defining consistent methods for people’s products to work together, ranging from simple cases of using a portable MP3 player to push songs for playback on a home stereo, to streaming video to and from a variety of network storage sources in multi-user scenarios, in personal and commercial configurations.

By providing guidelines for low-level technical interaction between electronic products, between client and server devices, DLNA seeks to ensure seamless compatibility and complementary service sharing across devices, giving manufacturers the flexibility to create their own custom interpretations, interfaces, and behaviors, while giving consumers the benefit of trouble-free usage in the home. To be entitled to stamp commercial products with the DLNA logo, a reassuring point of market recognition

for consumers, companies must go through a certification process. This process is greatly simplified for customers using pre-certified DLNA compatible solutions, such as those offered in mediaCTRL SDK solutions.

The DLNA guidelines cover the entire data chain between connected devices, similarly to the traditional OSI protocol layer model, ranging from low-level communications to high-level interface issues. Though these layers are handled (and thus greatly facilitated) by the provided mediaCTRL SDKs and their APIs, the diagram below shows the various elements that are part of creating compatible connected multi-media devices:

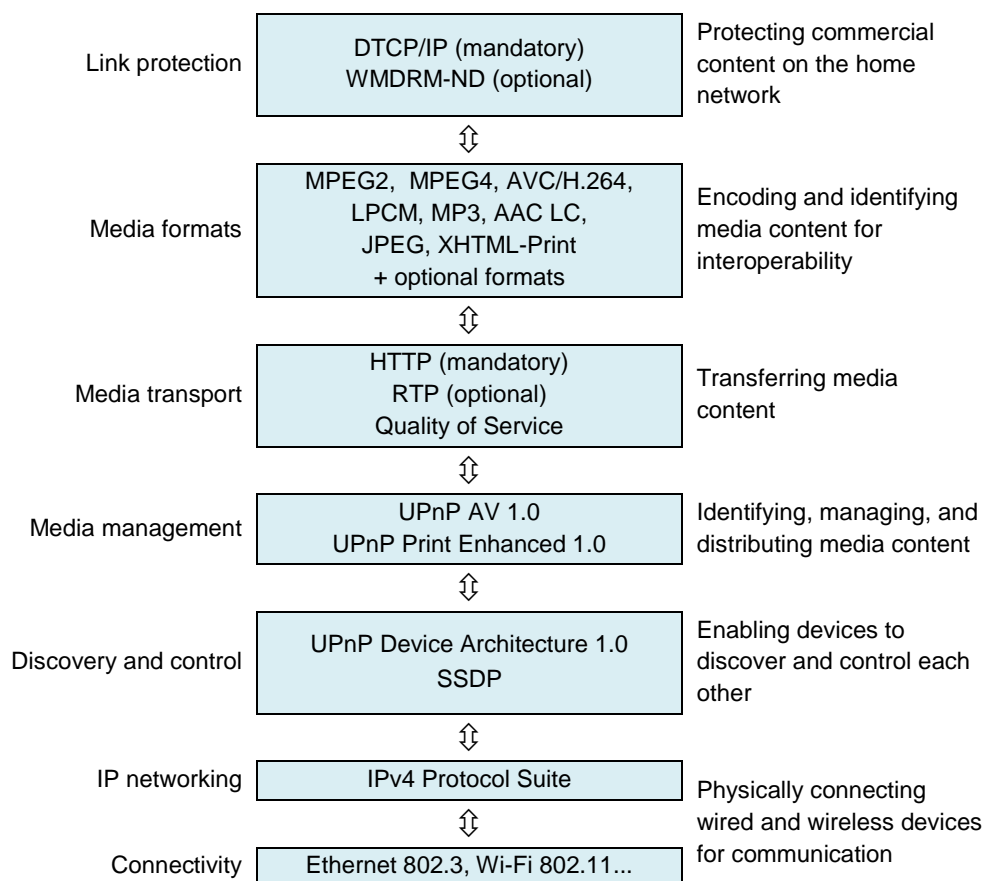


Figure 1 - DLNA multimedia handling layers

## Available AwoX mediaCTRL SDKs

Here is a summary of the main mediaCTRL SDKs. A complete list of current SDKs and plug-ins is available on [www.awox.com](http://www.awox.com).

AwoX mediaCTRL SDK	Application
Controller SDK	For client applications seeking to discover and control network servers and renderers
Player SDK	<i>Subset</i> of Controller SDK, for creating multimedia player devices
Server Plug-in SDK	For server-side application that publish multimedia files to client applications
Renderer SDK	For media playback devices, a full-featured multimedia renderer to play network content and access IPTV-type services
DTCP Source SDK	Server side link protection based on Digital Transmission Content Protection
DTCP Sink SDK	Client side link protection based on Digital Transmission Content Protection

*Table 1- mediaCTRL SDK summary*

In the next section, these SDKs are described in the context of their corresponding use cases.

## DLNA use cases and the AwoX offer

AwoX mediaCTRL SDK solutions are based on providing the features that customers may need to implement the various DLNA-defined use cases. These use cases focus on many different types of consumer electronic, entertainment, computing, and communication devices, such as:

- Televisions
- Personal computers and tablets
- Audio-video receivers
- Blu-ray and DVD players
- Network-attached storage
- Mobile devices
- Set-top boxes
- Printers
- Cameras and camcorders
- Digital photo frames
- Internet gateways and routers
- Game consoles
- Intelligent home appliances

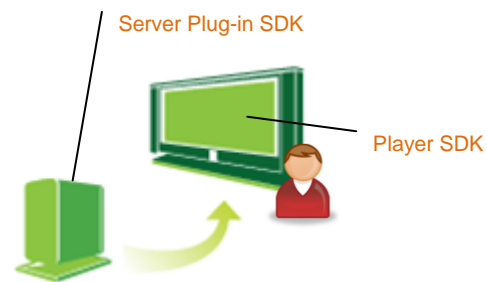
Generally speaking, these are the types of devices that make up today's digital home. These devices all exhibit different levels of processing power, storage capacity, interactive interfaces, etc., yet they can leverage the widely accepted industry guidelines to communicate with each other as appropriate, and in a predictable manner.

The next sections provide a description of the main DLNA use cases, along with the corresponding AwoX mediaCTRL SDK applicable to creating the scenario. Note that mediaCTRL SDKs are designed flexibly to meet the criteria for multiple usage scenarios.

## Media use cases

### Watch home movies

Store your movie collection on a Network-Attached Storage device (NAS) – a certified Digital Media Server (DMS). Instead of watching a movie on the small computer screen in your office, you can enjoy it on the large flat-panel TV in your living room. With a DLNA-certified TV as a Digital Media Player (DMP), you can use the DMP capabilities of your TV to find the movies on your NAS and play them on the TV.

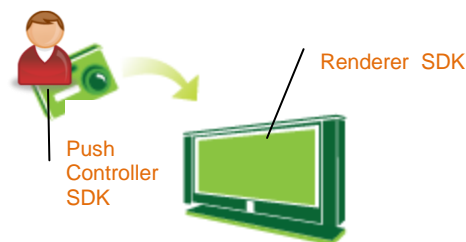


*Figure 2 – Use your TV to watch movies stored on a computer or network-attached storage device*

### Display photos

In this scenario, photos are stored on your digital camera – a certified Digital Media Controller (DMC) and you may choose to look at them on your TV. With a DLNA-certified camera, you can send the photos to a DLNA-certified TV.

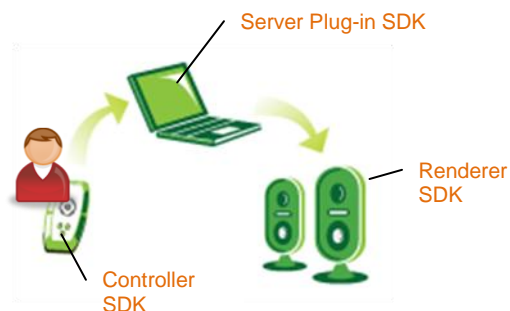
Assuming the TV is certified as a Digital Media Renderer (DMR), that capability is used to display the photos.



*Figure 3 – Display your camera's photos on an external screen*

### Listen to music

Load all your favorite music onto your computer – a certified Digital Media Server (DMS). Use your DLNA-certified smartphone or tablet, possibly also certified as a Mobile Digital Media Controller (M-DMC), to locate songs you want on your computer and send them to be played on your DLNA-certified wireless speakers. In this case, the speakers provide the capabilities of a certified Digital Media Renderer (DMR).



*Figure 4 – Control music playback without having to sit at your computer*

## Storage and printer use cases

### Back up pictures from your phone

With photos on your mobile phone, a certified Mobile Digital Media Uploader (M-DMU), you can send and save to your DLNA-certified Network Attached Storage device (NAS). Using a DLNA-certified mobile phone, it's easy to upload photos to your NAS operating as a certified Digital Media Server (DMS).

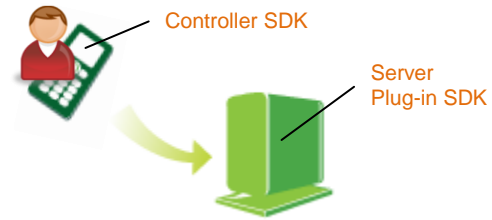


Figure 5 – Back up pictures from your smartphone

### Enjoy music on-the-go

In this scenario, you can transfer music from your PC, a certified Digital Media Server (DMS), to your mobile phone or player, a certified Mobile Digital Media Downloader (M-DMD). With DLNA-certified devices, the music can be downloaded by the mobile device's MP3 player using the PC as the source.

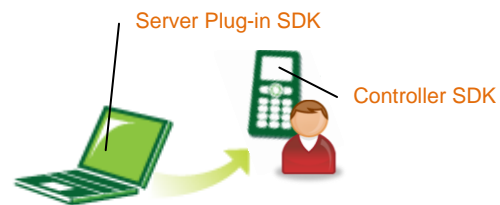


Figure 6 – Take your music on-the-go

### Print your photos

Mobile phones are increasingly used as carry-around cameras. To print photos from a phone that is certified as a Mobile Digital Media Controller (M-DMC), simply send photos to a printer that implements a certified Digital Media Printer (DMPr).

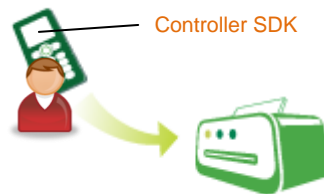


Figure 7 – Print pictures from a mobile device

## Internet and premium use cases

### Push Internet content to your TV

Simplify Internet and multimedia content search on your TV using a smart phone or tablet certified as a Mobile Digital Media Controller (M-DMC). Browse and push the content to your DLNA-certified TV operating as a certified Digital Media Renderer (DMR).



Figure 8 – Pushing Internet content to your TV

### Remote access

You can use your mobile phone to access and play content that is stored at your home or another location with a mobile phone that is certified as a Mobile Digital Media Player (M-DMP) with Remote Access capabilities, you can connect to a service portal, through which you may access your content remotely, such as that which is stored on a set-top-box equipped running a DLNA-certified server.

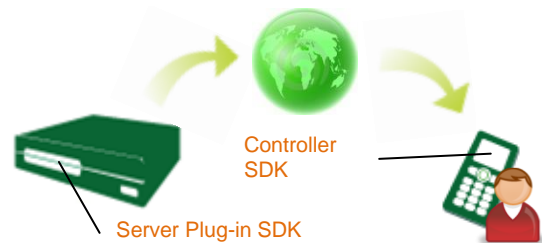


Figure 9 – Access remote content with your smartphone

### Link-protected content

A TV with a DLNA-certified Link Protected Digital Media Player (DMP) can play content from a DLNA-certified protected Digital Media Server (DMS).

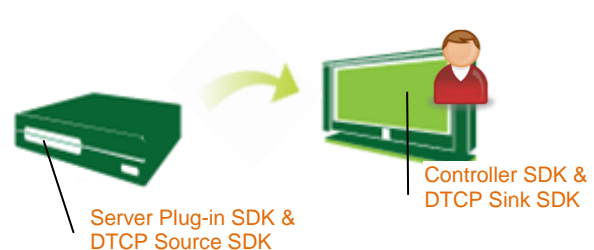


Figure 10 – Play content from a protected server

## Advanced use cases

### “Follow me”

While watching movie in your living room on a DLNA-certified TV equipped with a Digital Media Renderer (DMR) function, you might wish to move to your bedroom and continue watching the movie. With a DLNA-certified mobile phone equipped with a certified Mobile Digital Media Controller (M-DMC), you can pause the movie and push it to your bedroom TV, assuming it is also equipped with a DMR function. You can then change rooms and resume the movie.

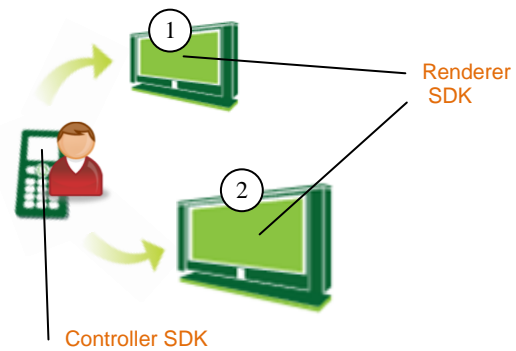


Figure 11 – Start watching a movie on one TV and finish it on another

### Remote User Interface

From your TV equipped with a DLNA-certified Remote User Interface Player (+RUIPL+), your customers can browse and play content using the identical user interface you provide on your gateway device, provided it includes Remote User Interface Source capability (+RUISRC+). Using this, coupled with “Link Protected Content” described earlier, reduces equipment costs and simplifies maintenance for providing service to multi-TV households.

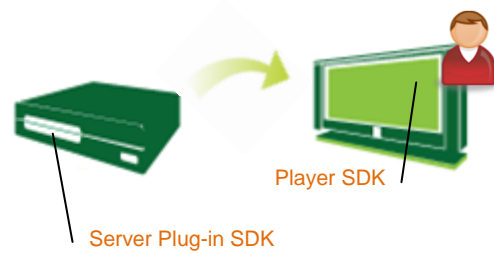


Figure 12 – Use your TV to watch content in an interface generated by your service provider

## Players and servers

### Digital Media Server (DMS)

Digital Media Server devices store content and make it available to networked Digital Media Players (DMP) and Digital Media Renderers (DMR). Some digital media servers can also help protect your content once stored. Examples: PCs and Network-Attached Storage (NAS) devices.



Figure 13 – Multimedia file storage on digital media server

### Mobile Digital Media Server (M-DMS)

This wireless device stores content and makes it available to wired or wireless-networked devices:

- Mobile Digital Media Players (M-DMP)
- Digital Media Renderers (DMR)
- Digital Media Printers (DMPr).

Examples: mobile phones, portable music players, digital cameras, and digital video cameras.

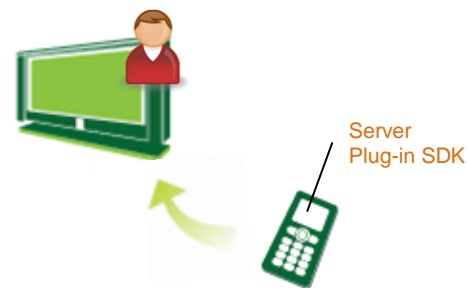


Figure 14 – Store content that can be accessed by other players and renderers

### Digital Media Player (DMP)

This device locates content on Digital Media Servers (DMS) and provides playback and rendering capabilities. Examples: TVs, Blu-Ray players, A/V receivers, and home theaters, and game consoles.



Figure 15 – Play content that is stored on other devices with server functionality

### Mobile Digital Media Player (M-DMP)

This wireless device locates and plays content on a Digital Media Server (DMS) or Mobile Digital Media Server (M-DMS).

Examples: mobile phones and mobile media tablets designed for viewing multimedia content.



Figure 16 – Use mobile devices to play

content stored on fixed and mobile servers

## Controllers and renderers

### Digital Media Controller (DMC)

These devices locate content stored on Digital Media Servers (DMS) and play it on Digital Media Renderers (DMR).

Example: computers.

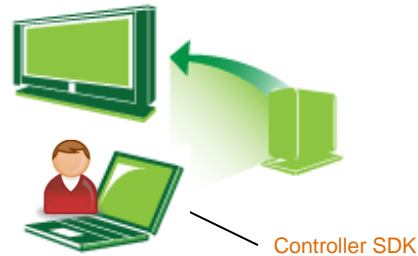


Figure 17 – Play content stored on a server device using a different renderer device

### Mobile Digital Media Controller (M-DMC)

These wireless devices find locate content on Digital Media Server (DMS) or Mobile Digital Media Servers (M-DMS) and send it to digital media renderers (DMR).

Examples: smartphones and tablets.

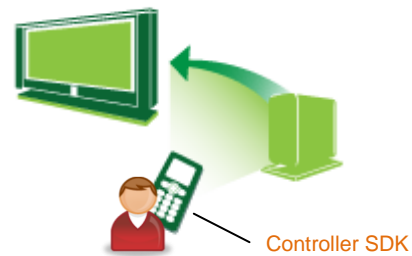


Figure 18 – Use a mobile device to play content stored on a server device using a different renderer device

### Digital Media Renderer (DMR)

These devices play content received from a digital media server (DMS), The DMS's content is located and routed to the DMR by a digital media controller (DMC).

Examples: TVs, audio/video receivers, digital radios and remote loudspeakers.

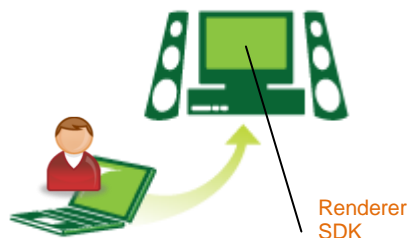


Figure 19 – Play content received from various controller devices

## Upload and download

### Mobile Digital Media Uploader (M-DMU)

These wireless devices send (upload) content to a digital media server (DMS) or mobile digital media server (M-DMS).

Examples: digital cameras and mobile phones.



*Figure 20 – Uploading content from mobile devices*

### Mobile Digital Media Downloader (M-DMD)

These wireless devices find and store (download) content from a digital media server (DMS) or mobile digital media server (M-DMS).

Examples: portable music players and mobile phones.



*Figure 21 – Downloading content onto mobile devices*

## DLNA device capabilities: upload, printer, Remote User Interface (RUI)

Feature	Description	AwoX SDK
Upload Controller (+UP+)	An Upload Controller uploads content from a remote server (DMS/M-DMS). The remote DLNA Server must run the Upload Device option.	AwoX mediaCTRL Controller SDK
Download Controller (+DN+)	A Download Controller downloads content from a remote DLNA server. No specific implementation is required on the server side.	AwoX mediaCTRL Controller SDK
Upload Device Option (UDO) for Server	Required feature for a server to be able to upload content.	AwoX mediaCTRL Server Plug-in SDK
Scheduled Recording Controller (+SR+) for Server	Implements Scheduled Recording Services and Control on a server.	AwoX mediaCTRL Server Plug-in SDK
3-Box Printer Controller (+PR2+)	A 3-Box Printer Controller also controls the printing of images to a printer (DMPPr), but does not actually serve it. It can browse and locate the image to be printed, as well as prepare the X-HTML Print Document.	AwoX mediaCTRL Server Plug-in SDK

Table 2- Additional DLNA functionality

## Conclusion

These use cases are based on the interconnection of compatible client-server solutions, combined with the appropriate rendering and playing services to enjoy music, images, and video in a variety of ways. The AwoX mediaCTRL SDK(s) that you use will depend on the device or solution that you intend to offer. Some devices, such as computers, may contain multiple DLNA-compatible applications. Others, such as cameras, may only contain one type of functionality (i.e. sending pictures).

Regardless of the target use case, mediaCTRL SDKs offer the clear advantage of providing you with a ready-to-use solution that not only abstracts away the challenges related to presenting multimedia content to users in numerous circumstances and on numerous types of devices, but it also handles complex communications, presentation, discovery, and access operations in a fully compatible DLNA world.

## mediaCTRL middleware architecture overview

The AwoX DLNA SDK offer is based on three main mediaCTRL kits.

- mediaCTRL Controller/Player SDK
- mediaCTRL Server Plug-in SDK
- mediaCTRL Renderer SDK

These SDKs offer numerous development possibilities for multimedia devices, and are frequently combined with complementary modules to add additional features. A single product may make use of several mediaCTRL SDKs in order to provide the desired overall functionality. Modules communicate within the final solution via S-Bus, a bus based on M-Bus for interprocess communications. The two diagrams below show examples of multiple components within a single physical product (S-Bus is described in more detail later in this chapter):

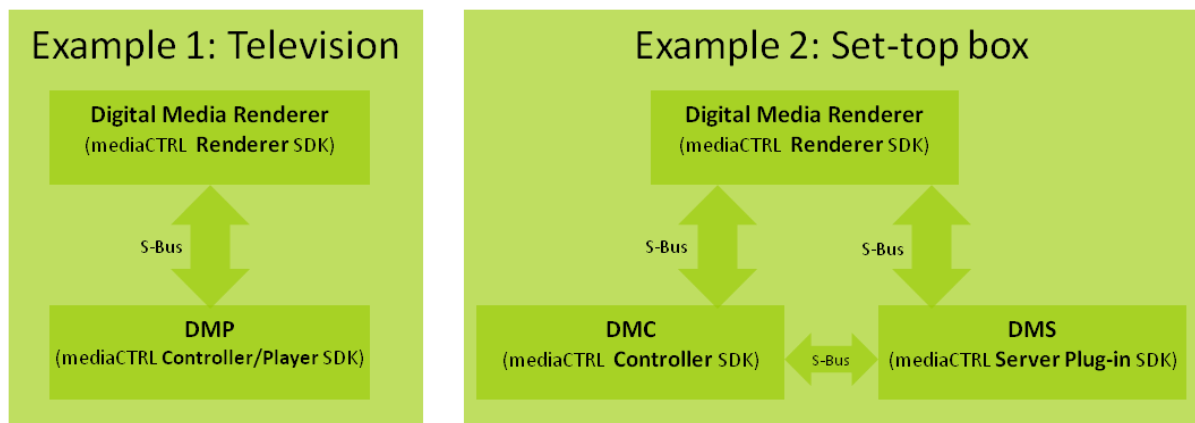


Figure 22 – Communication between software components takes place via S-Bus

The following sections provide an overview of these software tools, APIs, and modules.

### mediaCTRL Controller/Player SDK

The AwoX mediaCTRL Controller SDK gives you all the tools you need to design multimedia applications on devices ranging from set-top boxes to mobile phones and countless other types of media products, enabling you to discover media stored in servers and control playback on renderers available on the network.

The mediaCTRL Player SDK is a subset of the mediaCTRL Controller/Player SDK, enabling you to implement a DLNA DMP profile. It is designed to be integrated within any user interface framework, from a Web browser to an LCD screen. To ensure industry-wide compatibility, the mediaCTRL Controller/Player SDK is fully compliant with the DMC profile (defined by DLNA August 2009 guidelines) and with the UPnP AV 1.1 Control Point profile.

The diagram below shows the modular structure of the mediaCTRL Controller/Player SDK:

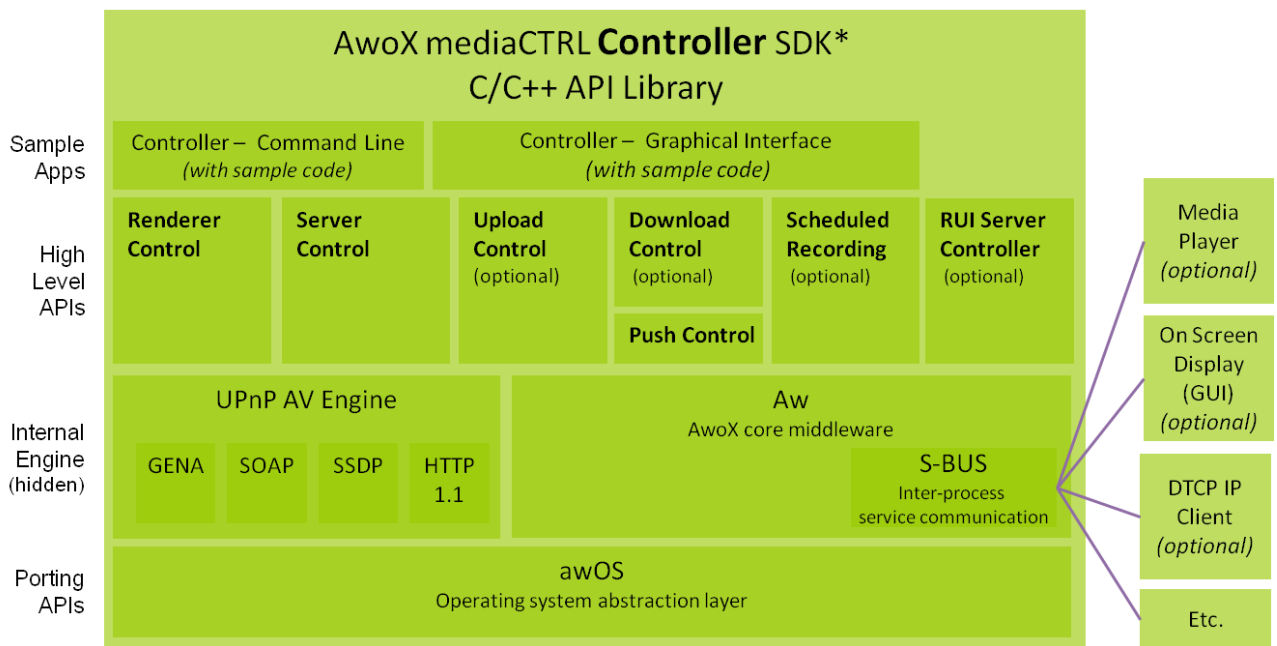


Figure 23 – Overview of the mediaCTRL Controller SDK

## High Level APIs

### Server Control



This module is provided both in the AwoX Controller SDK and AwoX Player SDK

Key features:

- Server discovery and removal
- Server event subscription and management
- Advanced browsing with caching
- Search support
- Thumbnail retrieval and caching
- Select best URI for the local player
- Automatic update (based on SystemUpdateID and ContainerUpdateID)

## Renderer Control

---



This module is provided in the AwoX Controller SDK only.

---

Key features:

- Renderer discovery and removal
- Renderer event subscription and management
- Select best URI for the active DMR
- Volume / Mute control
- Play, pause, resume, stop support
- Optimized position computation
- Seek (setting position in file)
- Play speed (such as x1, x2, x1/10, etc)
- SetNextURI for playlist mode

## APIs for optional components

### Push Controller (optional)

Key features:

- Select best URI for the active DMR
- Built-in minimal DLNA server
- Streaming with byte seek support
- Metadata extraction
- DLNA profile computation

### Download controller (optional)

Key features:

- Suspend / Resume download
- Preserve metadata
- Multiple download simultaneously
- Compute file name and extension from metadata and profile info

## Upload controller (optional)

Key features:

- Upload any container
- Create item
- Create container
- Delete item / container
- Suspend / Resume upload
- Multiple upload simultaneously

## Controller SDK performance indicators

- 1 thread for the UPnP stack
- 1 or more threads for the UI
- Code size (without UI): 250 kB + (600 kB shared)
- Code size (with UI) : 1200 kB + (1600 kB shared)
- Idle CPU < 1%
- Most CPU used by UI
- Command line and graphical front-end tools are both available

## mediaCTRL Server Plug-in SDK

The mediaCTRL Server Plug-in SDK provides a high-level C++ application programming interface so you can develop your own content services, ranging from broadcast and live source connectors to CD, DVD, removable mass storage, Internet bookmark applications, and more. Whether you are designing a networked hard drive, personal video recorder or some other kind of device that hosts mass media storage, AwoX Server Plug-in SDK not only serves media content, but it also gives you a simple way to develop your own services in compliance with DLNA August 2009 Guidelines.

The AwoX mediaCTRL Server Plug-in SDK is fully compliant with the DLNA 1.5 DMS guideline profile and UPnP A/V 1.1 Media Server profile.

The diagram below shows the modular structure of the mediaCTRL Server Plug-in SDK:

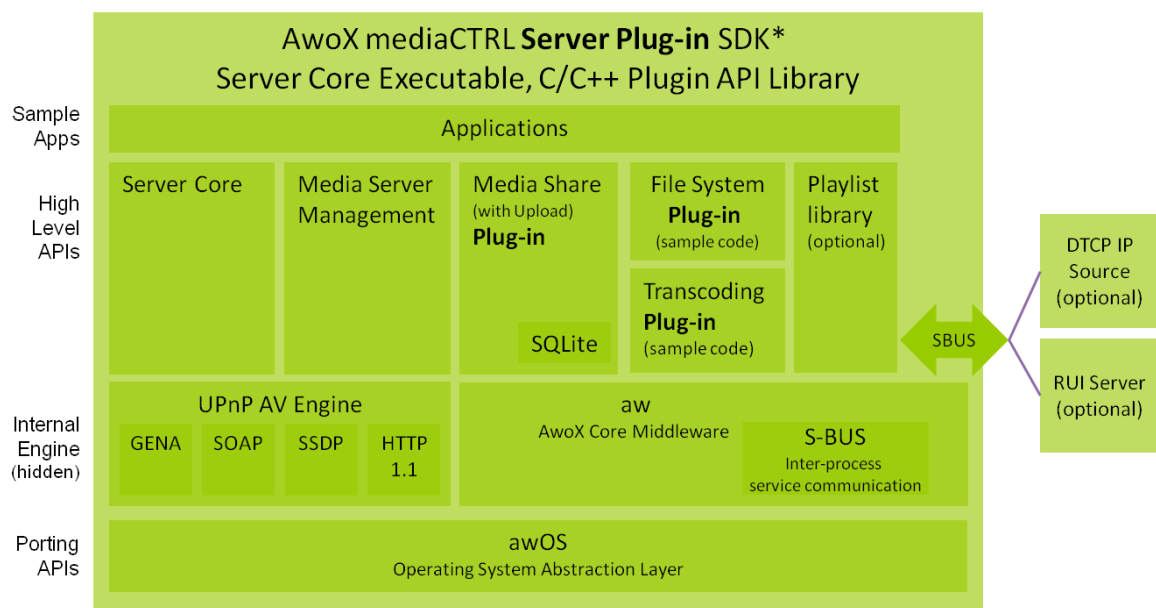


Figure 24 – Overview of the mediaCTRL Server Plug-in SDK

## How the Server Plug-in works

The mediaCTRL Server Plug-in architecture is based on event properties. That is, the Server Core shares a property list with a given plug-in and posts events to request the plug-in to perform specific actions. The Server Core can call C functions in plug-ins directly, such as new, setup, execute, terminate, etc.

Plug-in features include:

- The ability to post events to request the Server Core to perform a particular action
- Directly calling specific C functions in the Server Core
- Extremely stable and well-documented C++ API
- Generic extensible mechanism to exchange events between Server Core and plug-ins
- Extensible (backward and forward compatibility)
- Generic code, for platform and operating system independence
- Easy to use with same paradigm for different types of plug-ins (renderers, servers, control-points, etc.)
- Bi-directional operation (the plug-in can also call-back the Server Core)
- Small code size adapted to embedded constraints

Plug-in examples:

<b>Internet Radio Plug-in</b>	Retrieves XML data from Internet Database server and translates this data into a UPnP hierarchy
<b>File system Plug-in</b>	Suitable for low resources platforms. No database needed, adapts well to removable storage.
<b>YouTube Plug-in</b>	Direct browsing of YouTube content
<b>Live TV Plug-in</b>	Can serve channels and content from a local attached DVB receiver
<b>Weather Plug-in</b>	Retrieves weather info and bitmaps from the provider and translates this data into a UPnP hierarchy

## High Level APIs

### Server Core

Key features:

- Runs as a background application
- Compliant with DLNA August 2009 & market ecosystem (XboX,...)
- Functions

- Administration
- Browsing and search management
- HTTP streaming
- RTP streaming
- On-the-fly playlist generation
- Support for seek, background, interactive, and streaming transfer modes

## Media server management API

- Start and stop media server
- Select which folders to share (if applicable, depending on installed plug-ins)
- Request database content update
- Request specific folder update (partial update)
- Get server state (stopped or running)
- Set and get various parameters

## Plug-in Framework

This C++ framework is used to develop new plug-ins to:

- Serve remote content or live content
- Extend transcoding capabilities

### MediaShare Plug-in

- Based on SQLite by default (replaceable by your own framework)
- Index and organize content using metadata information (ID3, WMV/WMA, JPEG tags...)
- Use SQLite database to consolidate metadata and provide fast browsing and searching
- Handle 100,000 files with very fast response time
- Manage playlists
- Manage upload operations

### Transcoding Plug-in (sample code)

- Resize still images
- Transcode image files to JPEG format

## Server Plug-in SDK performance indicators

- 8 threads + 1 or more per plug-in
- Code size: 330 kB + (600 kB shared) + Plug-ins
- Idle CPU < 1%
- Browse request < 2% CPU usage
- Per stream < 3% CPU usage
- i-notify capable (or equivalent): instant refresh
- Index 20,000 files: 3 minutes
- Browsing speed for 20,000 file database from PS3: 3 to 4 seconds
- Index size: 25 MB for 60,000 Media Files

## mediaCTRL Renderer SDK

Whatever kind of media playback device you need to design, from television sets to set-top boxes to stereo systems and more, AwoX Media Render Plug-in SDK gives you both the ability to play network content and to extend rendering to other sources, such as IPTV services.

To ensure industry-wide compatibility, the AwoX Media Renderer Plug-in SDK is fully compliant with the DMR profile (defined by DLNA August 2009 guidelines) and UPnP AV 1.1.

The diagram below shows the modular structure of the mediaCTRL Renderer Plug-in SDK:

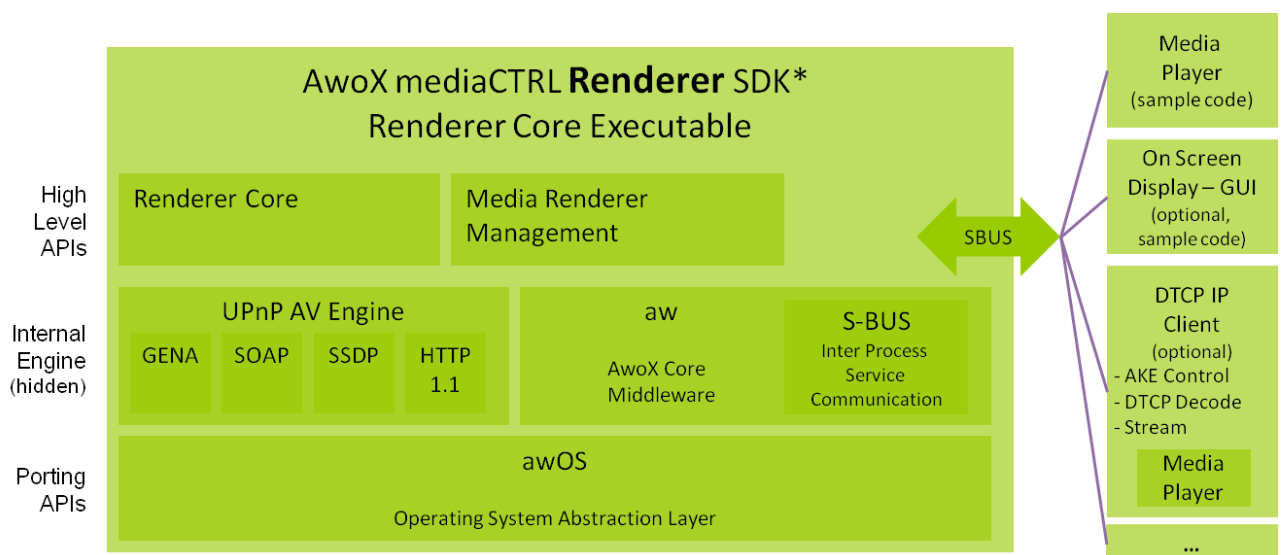


Figure 25 – Overview of the mediaCTRL Renderer SDK

## High Level APIs

### Renderer Core

- Runs as a background application
- Full featured media renderer
  - Compliant with DLNA August 2009 & market ecosystem (XboX,...)
  - Handles maintenance/update of renderer state variables: current URI, play time, play state, metadata, ...
- Provides HTTP (and RTP) streaming sink
- Supports multiple players
- Picture player displays most image formats: JPEG, PNG, BMP, GIF, TIFF
- Sample picture player included
- Framework to easily integrate new media players
  - Player plug-ins can be directly loaded by the DMR
  - Player can be running in a separate process (generic MBus plug-in provided for this)
- Includes configuration/settings API

### Media player sample code

- Media player provided as sample implementation of renderer SDK.
- Control API called by mediaCTRL Renderer SDK, to be customized for media player specifics
- This player directly supports<sup>1</sup>:
  - Helix (Linux & Win32)
  - gStreamer (Linux & Win32)
  - MPlayer (Linux and Win32)
  - DirectShow (Win32, Windows Mobile and Windows embedded)
  - STMicro MediaPlayer 1.0 & 2.0 on ST710x platforms
  - Ingenient & Ittiam media player on TI platforms
- State reporter provided
  - StateAPI called by media player

### Renderer SDK performance indicators

- 6 threads

---

<sup>1</sup> Please contact AwoX for details on other target platforms if necessary.

- Code size : 400 kB + (600 kB shared)
- Idle CPU < 1%
- Playback CPU load depends on AV back-end

## mediaCTRL DTCP SDK

### What is DTCP

Digital Transmission Content Protection (DTCP) is a means of protecting digital multimedia content as it transits along network connections. It is a low-level application that is completely transparent to end-users, but it enables providers to protect the content that is streamed from a digital media source (DMS) to compatible player devices (DMP), or digital media renderers (DMR). DTCP is considered “Link Protection” and is not a form of Digital Rights Management (DRM) or a Conditional Access System (CAS).

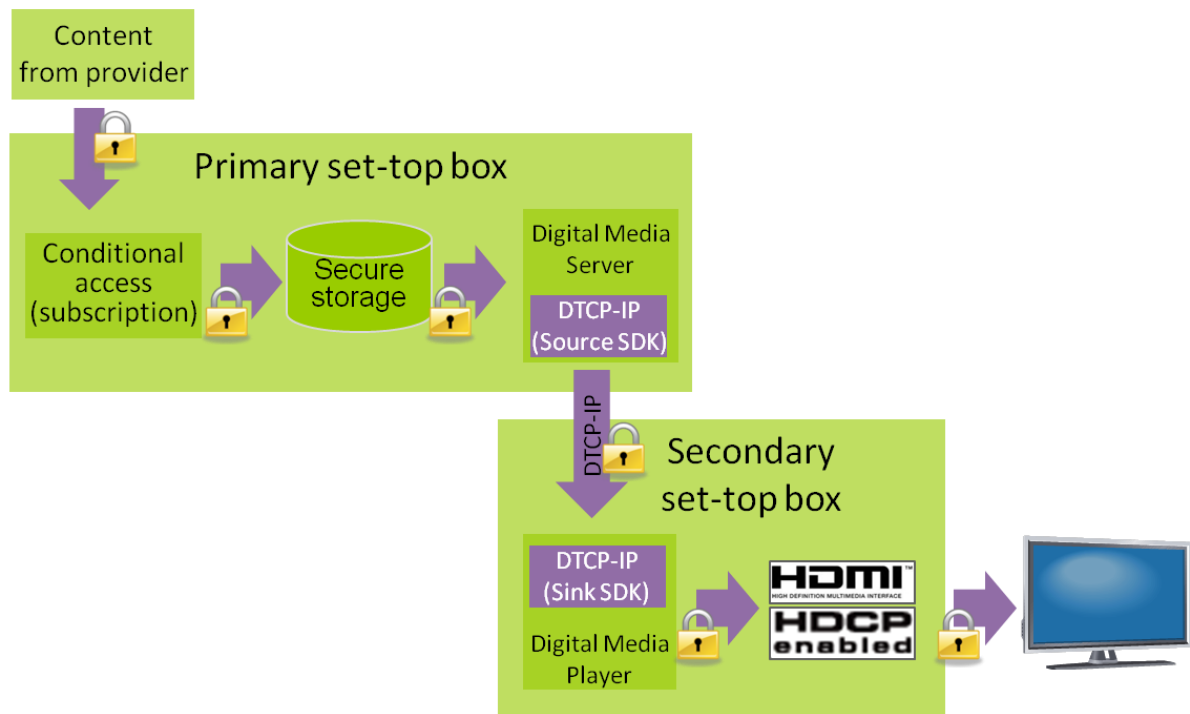
### Overview of mediaCTRL DTCP Source and Sink SDKs

The AwoX mediaCTRL SDKs enable developers to integrate this complex DTCP technology into their applications without having to deal with the numerous related challenges, including handling complex security issues and key exchange.

AwoX offers two separate DTCP SDKs with server (“*source*”) and player (“*sink*”) components to provide secure multimedia streaming between devices.

- DTCP Source SDK is used to integrate features to encrypt protected content to serve.
- DTCP Sink SDK is used to integrate features to decrypt protected content to display or play.

The DTCP Sink and Source components communicate over the network using TCP/IP. The diagram below shows an example in which DTCP provides a secure link over TCP/IP from one set-top-box to another. In this case, the set-top box on the left accesses the secure content, stores it locally, and then is able to serve it securely to another viewing device in the home:



*Figure 26 – Typical DCTP usage scenario for playing recorded video with active data protection*

Each component in the chain provides security for its own channel, thus providing 100% security for multimedia content from one end to the other.

Key features of the DTCP SDK include:

- **Copy control information (CCI):** enables content owners to specify how their content can be used (such as “copy-never”).
- **Device authentication and key exchange (AKE):** before sharing valuable information, a connected device must first verify that another connected device is authentic.
- **System renewability:** ensures long-term integrity of the system through the revocation of compromised devices.

## DCTP Source-side architecture and integration

From the source-side (server) perspective, the content is encrypted and then transmitted over a DTCP link, secured through automated authenticated key exchanges (AKE), as shown here:

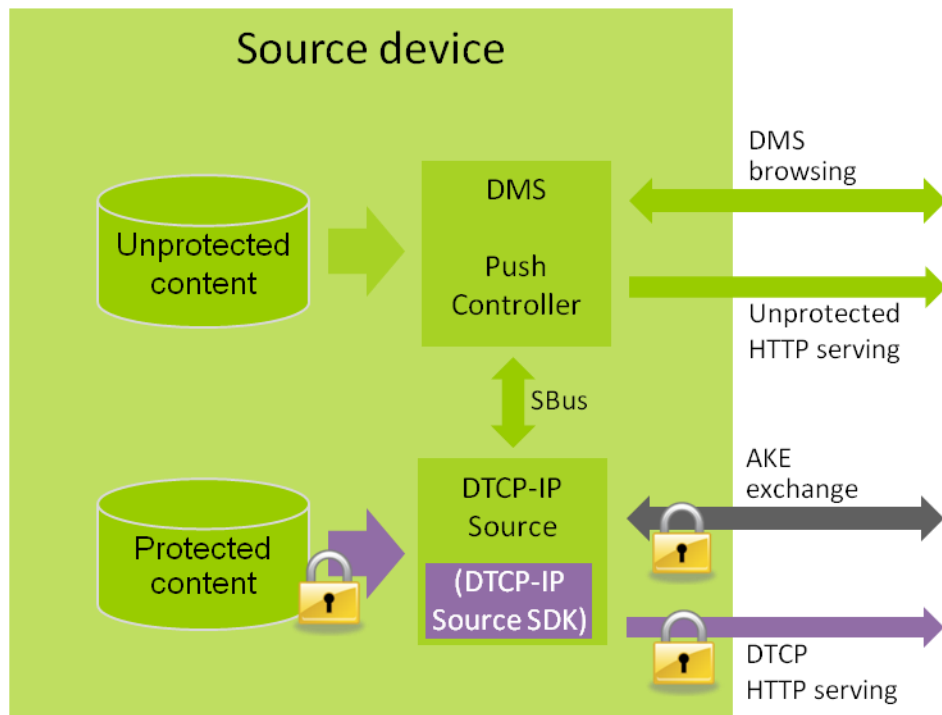


Figure 27 – Content is encrypted via DTCP and served over TCP/IP using keys

### Server-side integration considerations

- The DTCP-IP source/server can be located in a separate process for security reasons.
- The DTCP-IP source/server has its own HTTP server.
- A special set of messages is exchanged between the Digital Media Server (DMS) and the DTCP Source in order for the DMS able to correctly present DTCP-protected content.
- The mediaCTRL DTCP Source SDK is compatible with the mediaCTRL Server and Push Control feature of the mediaCTRL Controller/Player SDKs.
- AwoX provides unsecured DMS sample code that works with DTCP-IP Source, but all secret assets and security implementations can only be handled by the final system integrator (it is also the integrator's responsibility to decrypt the protected content in the first place in order to serve it).

## DCTP Sink-side architecture and integration

From the sink-side (renderer) perspective, the protected content is received over a DTCP link, secured through automated authenticated key exchanges (AKE), as shown here:

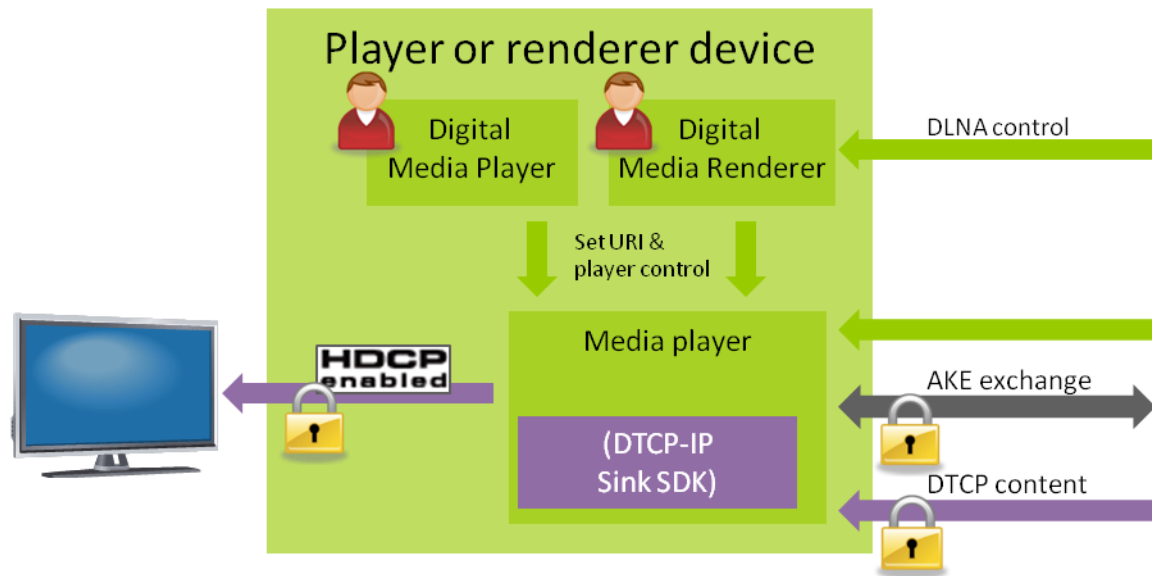


Figure 28 – The use of secure content via DTCP is completely transparent to end-users

### Sink-side integration considerations

- DTCP-IP sink must be integrated in the media player (security considerations are the integrator's responsibility, including ensuring HDCP security over HDMI).
- For DTCP-protected content, the player must use the streaming API in the DTCP-IP sink module instead of performing regular streaming.
- AwoX provides unsecured player/renderer sample code that works with DTCP-IP Sink, but all secret assets and security implementations must be handled by the final system integrator.

## DTCP certification

It is important to note that to be able to use DTCP with other compatible, secure products, the developing company must be a member of the DTLA (Digital Transmission Licensing Administrator) association. If your product containing DTCP security is going to be certified by DLNA in order to bear the DLNA logo, then you must run the product through special DLNA-DCTP testing.



AwoX is a member of DTLA and is thus authorized to deliver DTCP-related SDKs, but only to other DTLA members. As mentioned elsewhere in this document, AwoX provides tools to make working with DTCP easier for its clients, but the developer, integrator, or manufacturing company assumes full liability with respect to final DTCP

---

 implementations.
 

---

## Using S-Bus for inter-SDK communication

As mentioned in the beginning of this chapter, any given multimedia solution can make use of multiple mediaCTRL SDKs to provide various complementary features. Software components running in separate processes can communicate with each other via S-Bus, a bus based on the M-Bus inter-process communication (IPC) standard (RFC 3259). The mediaCTRL S-Bus module is integrated as a plug-in.

For example, in the diagram below, the renderer and player modules are running in different processes, communicating with each other via S-Bus.

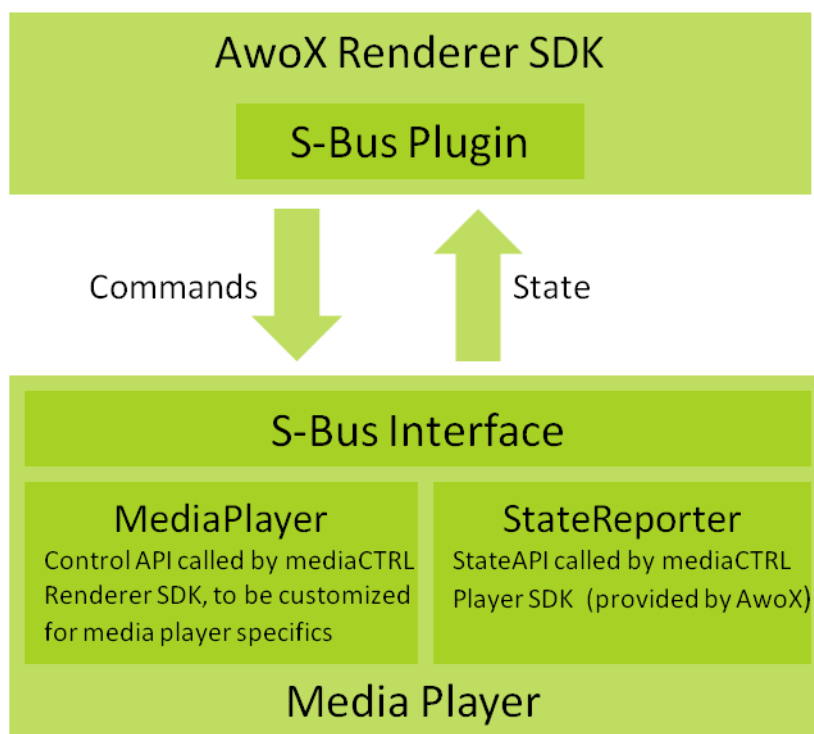


Figure 29 – Sample inter-process S-Bus communication between player and renderer

With support for UDP multi-cast, S-Bus provides a simple and flexible message-oriented communication channel among multiple applications on the same host machine.

There are two parts<sup>2</sup> to the S-Bus plug-in, communicating with each other via S-Bus:

- The first part runs in the renderer process
- The second part runs in the player process (AwoX provides a sample player with reusable code)

The advantage of this solution is that it hides the details of S-Bus communication, making your inter-process development significantly easier.

---

<sup>2</sup> The first part and most of the second part are implemented by AwoX.



The provided media player sample directly supports Helix (Linux & Win32), gStreamer (Linux & Win32), MPlayer (Linux and Win32), DirectShow (Win32, Windows Mobile and Windows embedded), STMicro MediaPlayer 1.0 & 2.0 on ST710x platforms, Ingenient & Ittiam media player on TI platforms.

---

## Working with AwoX

### What is included in the AwoX mediaCTRL SDKs?

The contents of the various mediaCTRL SDKs vary with the type of development you are targeting, as well as the hardware platform, operating system, and build environment. Precise details on SDK contents are provided in each delivery package.

Generally speaking, the following components are included:

- AwoX build framework
- Header files to include in custom code, or Java archive for Android targets
- Binary AwoX DLNA stack files for your target
- Sample Application with C++ source code, or Java source code for Android targets
- Complete documentation is provided for all mediaCTRL SDK in both PDF and HTML formats

Additional binary utilities may be required by the AwoX DLNA stack for your target (such as Python for development using MS Windows). In general, AwoX SDKs do not require any additional tools.

For more details or more specific information, please contact your AwoX representative.

### Target operating system requirements

AwoX SDKs support many different types of computer hardware and environments. The basic requirements for using mediaCTRL SDKs are as follows:

- GCC toolchain with support for C++, or a GCC-option compliant toolchain
- POSIX operating system, including Pthreads API
- BSD socket API

Constraints related to DTCP IP development:

- The DTCP client must be embedded directly within the final media player engine.



If the end-integrator is unable to build this media player, then it is not possible to integrate the AwoX DTCP-IP solution.

---

- If mediaCTRL DTCP SDKs are used in combination with DLNA-compatible modules, then the integrator must use the AwoX DLNA stack and SDKs.

## Getting started with your project

This section provides a quick summary of the process of getting started with AwoX on using mediaCTRL SDKs.

1. It is important to remember that there are numerous hardware platforms being used by consumer electronics manufacturers today. For this reason, the first action is for the developer to provide AwoX with a reference platform and appropriate toolchain.
2. AwoX proceeds with an evaluation of the platform and tools in order to assemble the appropriate package or, if necessary, consider porting the mediaCTRL to a new platform.
3. AwoX delivers the desired SDK, complete with the required APIs, binary files, and documentation.
4. The integrator client then installs the mediaCTRL tools into the target environment to pursue development.
5. AwoX provides training, support, and testing services as appropriate.

## Frequently Asked Questions

### **What is the difference between mobile device classes (prefixed with an M) and other profiles?**

Mobile device classes, named Mobile and Handheld Devices (MHD), are only different by the set of mandatory media format profiles that they support. For example, Home Network Devices (HND) with image, audio and video media classes must support MPEG2 (in PS & TS containers), LPCM, and JPEG. On the other hand, an MHD must support AVC (in an MP4 container), MP3, AAC (MP4 & ADTS containers), and JPEG.

### **What is the commercial video profile?**

The commercial video profile will be an extension to the DLNA guideline directed at serving premium content inside the home. This initiative is supported by several US, and European Service Providers.

### **What is the difference between US & EU commercial video profiles?**

The first phase of commercial video profile for Northern America (CVP-NA-1) focuses on delivering protected content inside the home, while CVP-EU-1 also delivers user interface, through Remote User Interface guidelines. These guidelines rely on a subset of CEA2014A. The media format adopted also varies between the two, focusing on regional specifics mostly related to differences in frame rate (25fps/30fps).

### **Is DLNA certification available for all device classes, all capabilities, and all media formats listed inside the DLNA guidelines?**

No, only a subset (listed below) is currently available for certification, although this is subject to evolution at any time:

<b>Device classes</b>	DMP, DMS, DMR, DMC, M-DMP, M-DMS, M-DMC, M-DMD, M-DMU, DMPr
<b>Device Capabilities</b>	+UP+, +DN+, UDO, +PR1+, +PR2+
<b>Media Format Profiles</b>	<p><u>Video</u>: MPEG_PS_NTSC, MPEG_PS_PAL, MPEG_TS_JP_T, MPEG_TS_SD_EU, MPEG_TS_SD_EU_ISO, MPEG_TS_SD_EU_T, MPEG_TS_SD_KO, MPEG_TS_SD_KO_ISO, MPEG_TS_SD_KO_T, MPEG_TS_SD_NA, MPEG_TS_SD_NA_ISO, MPEG_TS_SD_NA_T, AVC_MP4_BL_CIF15_AAC_520, MPEG_TS_HD_KO_ISO, MPEG_TS_HD_KO_T, MPEG_TS_HD_NA, MPEG_TS_HD_NA_ISO, MPEG_TS_HD_NA_T, WMVHIGH_FULL; WMVMED_BASE, WMVMED_FULL, WMVMED_PRO</p> <p><u>Audio</u>: LPCM, AAC_ADTS_320, AAC_ISO_320, MP3, WMABASE, WMAFULL, WMAPRO</p> <p><u>Image</u>: JPEG_SM, JPEG_LRG, JPEG_MED, PNG_LRG</p>

## Certification and testing

One of the predominant reasons for using mediaCTRL SDKs as a tool to accelerate and facilitate creating multimedia applications is to create products that are compatible in the connected DLNA world. When DLNA compatible products from different vendors are used together, they work as a single coherent product comprised of multiple components. Consumers thus see the DLNA label as a guarantee of interoperability between brands and products, each one enhancing the performance of the other in a way that is natural and transparent in operation.

The most visible proof of this compatibility is the DLNA logo. This logo is typically printed on products and product packaging where consumers can identify it easily when considering product purchases. In order to be able to use the DLNA logo, you must run your product through a certification and testing process.



*Figure 30 – Certification is required in order to use the DLNA logo on your products*

The basic premise of DLNA certification is to prove compliance and interoperability.

### Compliance + Interoperability = Certification

<b>Compliance</b>	Products must pass tests to demonstrate that DLNA-compatible components indeed implement all of the required features for a given device class.
<b>Interoperability</b>	Products must interact smoothly with products from other vendors, as expected.

### Clearly defining your product profile

One of the most helpful things you can do when undertaking development that is DLNA compatible, and is to be certified as such, is to clearly define the profile and feature set of your product or solution. We highly recommend that you start with this seemingly obvious task as early as possible in order to make sure that you are developing efficiently.

As DLNA is a wide open world of multimedia, service-oriented, and technical features, it is very important to define the subset of features that you intend to support, including:

- For DMS (Digital Media Server), elaborate a clear definition of target media formats, and features for sorting, searching, uploading, seeking, etc.
- For DMC (Digital Media Controller), clearly define features for searching, uploading, downloading, etc.
- For DMR (Digital Media Renderers), plan ahead for supported media formats, seek, pause, etc.

Device profiles are particularly applicable for use with the Conformance Test Tool (*described below*).

You must also take care to clearly define the supported hardware devices:

- Reference devices are available for compatibility testing
- Preferred devices may be defined for specific compatibility requirements

The above definitions will in turn help you elaborate a clear and rigorous plan for Quality Assurance, including setting up the appropriate validation environment and test plan.

## Available test resources

A number of resources are available to developers, in particular the Conformance Test Tool (CTT) and Media Capability Verification Tool (MCVT), both of which are provided by DLNA to its members. Integrators generally leverage these tools, as well as Quality Assurance and pre-testing services from Independent DLNA Certification Vendors.

### **Conformance Test Tool (CTT)**

Based on DLNA Interoperability Guidelines, this tool provides protocol-level tests for DLNA-compatible devices. The CTT is updated about every six months, as well as after guideline updates and bug fixes.

### **Media Capabilities Verification Tool (MCVT)**

This tool defines the reference media files (Content Test Set) that can be used to check the robustness and reliability of DLNA-compatible solutions.

### **Content Test Set**

DLNA provides royalty-free and legally licensed multimedia content for developing and testing products in compliance-based environments. This test set includes several hundred media files, completely covering mandatory profiles, and partially covering optional profiles. DLNA updates these files regularly, generally every 6 months.

### **Link Protection Test Tool (LPTT)**

Used for testing DTCP-IP and WM DRM features. AwoX provides updates following all major changes and when tools are added.

## Plug-Fests

In addition to the above technical tools, many vendors partake in Plug-Fests, regularly organized at different locations around the world. Plug-Fests are not actually part of the certification program, but they offer valuable platform for testing products with thirty or more products from other vendors in a managed environment based on usage scenarios. This represents a valuable and cost-effective opportunity to test new products before their release, and before submitting them for official certification testing. Many people consider that it is a good sign to do well at Plug-Fests. In addition, the reference devices that are used in certification are on-hand for informal testing.



AwoX is a major provider of reference devices used for DLNA certification. Contact your AwoX representative for more information on obtaining these platforms for your own in-house use.

---

## UPnP certification

UPnP certification is not part of DLNA testing, but for DLNA device classes that function as UPnP devices, you are required to have a UPnP Certification ID (provided by the UIC) in order to submit the application for DLNA certification.

- For v1.0 certification → DMS only
- For v1.5 Certification → DMS, M-DMS, DMPPr, DMR<sup>3</sup>

Your company must be a UPnP Forum *Implementer* member and submit a UPnP Cert ID to DLNA *prior* to taking the Independent Certification Vendor test, to demonstrate that your product software has passed the UPnP test.

## Wi-Fi certification

If you are creating a product that offers IEEE 802.11 Wi-Fi functionality, you must first obtain Wi-Fi certification before submitting the product for DLNA certification.

For IEEE 802.11a/b/g/n, including WPA2 Wi-Fi (Protected Access), either WPA2-Personal or WPA2-Enterprise certification is accepted. To obtain Wi-Fi certification, your company must be a member of the Wi-Fi Alliance.

---

<sup>3</sup> Check for latest information on applicable device classes.

## Working with Independent Certification Vendors

When your product or solution is ready for final certification, you must contact an authorized DLNA test facility. These Independent Certification Vendors (ICVs) handle that actual certification process.

ICVs generally provide pre-testing services as well as final testing, to help ensure that your products pass the process with success. Among other things, ICVs are experts in running the Conformance Test Tool and benefit from having all the required equipment and Test Bed References devices to ensure interoperability. As mentioned earlier, AwoX reference platforms are officially used and recommended by DLNA. These platforms are highly useful to you as a tool during your development efforts and are available from AwoX.

Please contact your AwoX representative for more information on the certification process.

## Appendix - mediaCTRL SDK details

### Parameters

#### SSDP configuration

<b>Mask</b>	SSDP matching mask (default "255.255.255.0"). If the device IP address combined with this mask matches, then the stack will process discovery of this device. If not, the device will not be discovered.
<b>Port</b>	SSDP port number (default 1900).
<b>TTL</b>	SSDP TTL, which is the maximum number of gateways and routers the SSDP multicast packets will go through (default 4, minimum 1, maximum 100).
<b>SearchRequestCount</b>	Number of search requests to send when searching for a device type (default 4, minimum 1, maximum 10).
<b>MSEARCHInfoAny</b>	If set to 1, control-point will receive CmdSearcherInfo commands to search for any device (default 0). This requires "EnableMSEARCHInfo" in the "ControlPoint" section to be active.
<b>MSEARCHInfoCacheDuration</b>	Cache duration in seconds for M-SEARCH requests. Caching requests avoid flooding the client with numerous CmdSearcherInfo commands (default 60, minimum 1, maximum 3600). This requires "EnableMSEARCHInfo" in the "ControlPoint" section to be active.

#### UPnP configuration

<b>Exclude-if</b>	Comma-separated list of Ethernet interfaces to exclude. See <code>server_configuration_if</code> .
<b>UPnPLocal</b>	Can be 0, 1, or 2. Set this to 0 (default value) for the server to run in localhost and network mode. Set this to 1 for the server run in localhost only mode (i.e. it will not be visible on the network). Set this to 2 to have all interfaces except localhost.
<b>MaxEventSubscribers</b>	Maximum number of event subscribers that this device will support (default 10, minimum 1, maximum 200).
<b>DisableInitByeBye</b>	Specifies to not send bye-bye SSDP messages when the device is started (default 0).
<b>UDN</b>	Device UDN
<b>PreviousUDN</b>	Previous UDN of the device (used for initial SSDP bye-bye message).

## ControlPoint configuration

**UserAgent** Specifies the "User-Agent" header line for HTTP requests. Note that additional user agent information will be appended to this header line.

## Web server configuration

**SessionIdleTimeOut** Time in seconds that pipelined HTTP connections are kept open (default 20, minimum 3, maximum 600).

**MaxConnections** Maximum number of supported incoming connections (default 10, minimum 1, maximum 100).

**PortNumber** Server port number (0 to get a dynamically allocated port).

**InitialBufferSize** Initial memory size allocated for the receive buffer (default 10kB, minimum 2kB, maximum 256kB).

**MaxReadAheadSize** Memory size allocated for reading data ahead when sending (default 10kB, minimum 0kB, maximum 256kB).

**ConnectTimeOut** HTTP connection timeout in seconds (default 10, minimum 1, maximum 600)

**DisconnectLingerDuration** Time in seconds that a socket will still try to send data after it has been closed (default 60, minimum 0, maximum 600).

**DestroyLingerDuration** Time in seconds that a socket will still try to send data after the module been destroyed (default 5, minimum 0, maximum 600).

## Web client configuration

**SessionIdleTimeOut** Time in seconds during which pipelined HTTP connections are kept open (default 3, minimum 20, maximum 600).

**MaxIdleSessions** Maximum number of sessions to keep in an idle state (default 20, minimum 5, maximum 100).

**SessionPoolSize** Maximum number of sockets that can be used for outgoing connections (default 10, minimum 1, maximum 100).

**ConnectRetryCount** Number of retries to attempt when a connection fails. The interval between connection attempts will increase over time (default 4, minimum 1, maximum 10).

**InitialBufferSize** Initial memory size allocated for the receive buffer (default 10 kB, minimum 2 kB, maximum 256 kB).

**MaxReadAheadSize** Memory size allocated to read data ahead when sending (default 10 kB, minimum 0 kB, maximum 256 kB).

**ConnectTimeOut** HTTP connection timeout in seconds (default 10, minimum 1, maximum 600).

**DisconnectLingerDuration** Time in seconds that a socket will still try to send data after it has been closed (default 60, minimum 0, maximum 600).

**DestroyLingerDuration**

Time in seconds that a socket will still try to send data after the module been destroyed (default 5, minimum 0, maximum 600).



## Porting requirements

- Porting material requirements
  - Reference platform and toolchain provided to AwoX
- Porting technical requirements
  - CPU: 200MHz GPP (ARM or equivalent)
  - gcc toolchain with g++ support
  - TCP/IP BSD API with support for multicast
  - Posix threads API is available (with support for mutex, semaphore and condition variable)
  - Posix file API is available
  - Posix memory management API: malloc/free/realloc is available
  - Media file to be shared should be available in a standard file system directory/hierarchy

## Device classes and supported operating systems

The table below provides a summary of device classes and operating systems supported by mediaCTRL SDKs:

Aspect	Description
<b>Supported Device Classes</b>	DLNA 1.5 DMP, DMS (with Upload Device Option & OCM), DMR, DMC (+UP+, +PU+, +DN+)
<b>Provided APIs</b>	DMC API (server and renderer management; upload, download, and push controller APIs) DMP API (server management; upload, download, and push controller APIs) DMS binaries (with configuration API) DMS plug-in development API DMR binaries (with configuration API) DMR media player API
<b>Linux Ports</b>	Linux 2.4 (LinuxMIPS for Broadcom BCM6348) Linux 2.6 (Montavista, Timesys, kernerl.org, STLinux, ELDK....)
<b>Microsoft Windows® ports</b>	Windows XP, Vista, 7, CE 5.0, CE 6.0 Windows Mobile 6.5
<b>RTOS ports</b>	eCos, ucLinux (SMP862X)
<b>Support for wrapper API's instead of direct system calls</b>	Provided by the AwoX awOS porting layer, basically abstract, memory, sockets, threading, synchronization, file system access
<b>Native programming languages</b>	C & C++ (no template, STL, or exception): 30% C and 70% C++ AwoX offers C, C#, Java and Javascript APIs for the mediaCTRL Controller SDK.
<b>OS requirements</b>	Subset of POSIX API Support for multithreading Shared object library (may be optional)
<b>Modules required on target environment</b>	Very little system dependencies. AwoX uses only C and C++ standard libraries. All other libraries are provided, but target system libraries (eg zLib) may be used if provided and ABI compatible
<b>Memory Analyzers</b>	Valgrind, Massif, Lint

*Table 3 - Device classes and supported operating systems*

Given the fast-evolving nature of the technology industry, this information is subject to change at any time. Please contact AwoX directly for the latest details.

## Supported platforms

The table below offers a non-exhaustive list of platforms supported by mediaCTRL SDKs

Processor	Reference board	Target OS	Host OS	Features
TI TMS320DM6446	DVEVM	MontaVista Linux Professional Edition 4.0	Linux PC	DMC,DMS,DMR,DMP
TI TMS320DM644X	AwoX SALAI	MontaVista Linux Professional Edition 4.0	Linux PC	DMC,DMS,DMR,DMP
TI TMS320DM365	DV	MontaVista Linux Professional Edition 4.0	Linux PC	DMC,DMS,DMR,DMP
STM STB710X	STB7100REF	STLinux 2.2	Linux PC	DMC,DMS,DMR,DMP
NXP STB220	STB220	ELDK 4.0	Linux PC	DMC
XScale PXA270	AwoX	Windows CE 5.0	Windows	DMC
AMCC 440EP	EP440CX EPXH	Linux 2.6.14	Linux PC	DMC,DMS
Atmel AT91SAM9260	AwoX MOZART	TimeSys Linux 2.6.19	Linux PC	DMC,DMR,DMP
Broadcom BCM6348	Inventel DV4210	LinuxMIPS 2.4.17	Linux PC	DMS
SMP8634/35	DTV8634 RevA1	Linux 2.6	Linux PC	DMC,DMS,DMR,DMP
SMP8654	MediaTank	Linux 2.6	Linux PC	DMC, DMS, DMR, DMP
Independent	HTC & Sony-Ericsson Phones	Windows Mobile 6.1 & 6.5	Windows	M-DMS, M-DMC
Independent	HTC Hero, Dream, Magic, Desire HD, PRo988 Samsung Galaxy S	Android 1.5 ,2.0, 2.1, 2.2,3,3.0,4.0	Linux PC	M-DMS, M-DMC, M-DMP
Proprietary	iPhone 3G & 3GS	iOS4	MacOS X	M-DMC
Intel xB6 or equivalent	PC	Windows XP/Fedora Core 3	Windows/Linux	DMC,DMS,DMR,DMP

Table 4 – Support platforms (non-exhaustive list)

## Stack details

The table below provides technical details regarding mediaCTRL stack operation.

Aspect	Description
<b>Memory usage configuration</b>	Handled through configuration files Detailed in SDK documentation
<b>Thread usage</b>	Controller is single thread Server is 7 threads + 1 or more per plug-in Renderer is 5 threads + 1 or more per plug-in
<b>Code size</b>	Controller: 250 kB + (700 kB shared) (without graphical user interface) Server: 400 kB + (700 kB shared) + Plug-ins Renderer: 500 kB + (700 kB shared) + Plug-ins
<b>Performance</b>	Controller has no impact; only depends on graphical user interface. Server: ~1%, Browse: ~2%, Per stream: ~3%, i-notify capable Renderer: no impact, only depends on media player
<b>Multiple interface</b>	Multiple interfaces can be supported simultaneously. The stack automatically detects the interface. It is possible to exclude one or more interfaces through a configuration file.
<b>Limitations on storage parameters</b>	There is no hard-coded limitation. Managed through a configuration file.
<b>Method followed for simultaneous actions</b>	Commands are serialized on a first-come first-served basis.
<b>Error logging &amp; tracing</b>	All code has built-in logs and traces. This can be configured at runtime and completely removed in release build. Debug and release version of binaries are provided.
<b>Playlist support</b>	DMR & DMS support M3U, WPL, ASX and PLS
<b>Multilingual support</b>	Handled with configuration files for each module. Localization is also supported through .po files
<b>Timeout configuration</b>	Some time-outs can be configured through the API, most configuration is handled through configuration files.
<b>Pre-fetch &amp; abort</b>	Each action has a default timeout of 30 seconds (per DLNA specifications) and is then aborted. Pre-fetch is performed when browsing long media lists.
<b>Latest Conformance Test Tool test</b>	DMR, DMC (w +UP+, +PU+, +DN), DMS (w Upload Device Option) have passed Conformance Test Tool version 1.5.00.56 with no failures (logs available upon request).

*Table 5 – mediaCTRL stack details*

## Streaming protocols

The table below lists the streaming protocols supported by mediaCTRL SDKs.

Aspect	Description
HTTP 1.0/1.1	Yes
RTP/RTSP	Yes
Persistent Connections/Pipelining	Yes
Trick mode supported	Byte-Base seek, time-based seek for some media (including WMDRM protected content)
Buffering	Buffering is performed. Buffer count and size are configurable. The overall stream buffer can be split into smaller memory blocks to avoid fragmentation.
Buffer size configurability	Handled through configuration files. Detailed in the SDK documentation
Mechanism for streaming control information	Asynchronous control via call-back API or IPC-like mechanism

*Table 6 – Streaming protocols supported by mediaCTRL*

## Media players

The table below describes media player features as provided by mediaCTRL SDKs.

Aspect	Description
<b>Rendering control</b>	Yes
<b>Connection manager</b>	Yes
<b>AV transport</b>	Yes
<b>Can DMC only be enabled?</b>	Yes
<b>Can DMR only be enabled?</b>	Yes
<b>Support for multiple instances</b>	Yes. There is no hard-coded limitation. This can be defined by the application.
<b>Metadata Info</b>	Yes. All metadata is made available (as long as it was requested in the Browse or Search request).
<b>Trick mode support</b>	Varies on the platform player capabilities, but if supported by the player: pause, seek, fast forward, fast rewind and play speed can be supported.
<b>Resource properties</b>	Multiple resource properties are supported, alternate (i.e. transcoded) and thumbnail resources are handled. Resource optimization (depending on the renderer capability) is also computed
<b>Media Player Support</b>	<p>The following media players have already been integrated with mediaCTRL:</p> <ul style="list-style-type: none"> <li>• MPlayer (Linux and Win32)</li> <li>• DirectShow (Win32, Windows Mobile, and Windows Embedded)</li> <li>• Helix player (Linux)</li> <li>• gStreamer (Linux)</li> <li>• STMediaPlayer (ST710x platforms)</li> <li>• Ittiam &amp; Ingenient player (Texas Instruments platforms)</li> </ul> <p>AwoX acquired significant experience while working on these different frameworks, using and optimizing the media players for different processors. This includes dealing with architectures not supporting floating-point architecture.</p>

*Table 7 – mediaCTRL media player details*

## Media server

The table below describes media server features as provided by mediaCTRL SDKs.

Aspect	Description
<b>Content directory service</b>	Yes
<b>Connection manager</b>	Yes
<b>Availability of database for DMS</b>	Yes
<b>Database features</b>	Playlist support, browsing by creator, genre, album, media type (audio, image, video) and by folder (etc...). Support for DLNA OCM (creation, deletion, etc..).
<b>How many simultaneous sessions?</b>	This can be configured through a configuration file.
<b>Transcoding support</b>	Transcoding can be performed by a specific renderer plug-in SDK. May depend on platform hardware capabilities.
<b>Metadata Info</b>	Images: JPEG, BMP, PNG, JPEG and GIF Audio: MP3, WMA, PCM, WAV, AAC, AC3,... Video: WMV, MPEG2, MPEG4, H264, DivX, xVid,... Containers: AVI, DIVX, MPEG-TS, MPEG-PS, MP4, 3GP, ... New parsers may be added to the mediaCTRL framework easily.

Table 8 – mediaCTRL media server details

## Glossary

**Audio/Video (AV)** is an acronym used for Audio with Video, referring to any media containing both sound and moving picture.

**CODEC** is a program capable of performing encoding and decoding on a digital data stream or signal. The word codec may be a combination of any of the following: 'Compressor-Decompressor', 'Coder-Decoder', or 'Compression/Decompression algorithm'.

**Consumer Electronics (CE)** are electronic equipment intended for use by everyday people. Some categories of consumer electronics include telephones, audio equipment, televisions, calculators, and video players and recorders.

**Control Point (CP)** In *UPnP*, is a program that retrieves UPnP service and device descriptions, sends actions to services, and receives events from services.

**Copy Protection** is any technical measure designed to prevent duplication of information. Copy protection is often emotionally debated, and is thought to sometimes infringe on some users' property rights: for example, the legal right to make a backup copy of a videotape they have purchased, to install and use computer software on multiple computers, or to upload their music into their digital audio player for easier access and listening.

**Device Control Protocol (DCP)** In *UPnP*, is a specification standardized by UPnP Forum. Each device is defined by a XML description and includes vendor-specific, manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific web sites, etc. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation. For each service, the description includes a list of the commands, or actions, to which the service responds, and parameters, or arguments, for each action; the description for a service also includes a list of variables; these variables model the state of the service at run time, and are described in terms of their data type, range, and event characteristics.

**Digital Living Network Alliance (DLNA)** is a standard body of more than 350 companies in the consumer electronics, mobile and personal computer industries (Motorola, Philips, Samsung, Matsushita, Hewlett-Packard, Sony, Microsoft, Intel and Nokia are the founding members.). Its aim is to align companies and have industry standards, which will allow products from all companies to be compatible with each other and to enable a network of electronic devices in the home. DLNA first released 1.0 requirement in 2004. DLNA 1.5 requirements were released in March 2006.

**DLNA certification** is the process by which products are tested and certified in order to be able to use the DLNA logo on products. As DLNA is an umbrella set of guidelines that encompasses UPnP functionality, DLNA certification also involves UPnP certification. The DLNA certification process is unrelated to other potentially required certifications for a given product, such as Bluetooth or Wi-Fi.

**Digital Media Adapter (DMA)** act as bridges between two worlds, retrieving media over a network from a PC, and playing it through the existing audio and video equipment. Digital Media Adapter belongs to the class of Digital Media Player in DLNA vocabulary.

**Digital Media Controller (DMC)** From *DLNA 1.5* is a device or program finding content exposed by a Digital Media Server and matching it to the rendering capabilities of a Digital Media Render and setting up the connections between the Digital Media Server and the Digital Media Renderer.

**Digital Media Player DLNA (DMP)** is a device or program finding content exposed by a Digital Media Server and rendering the content locally. See below use case defined by DLNA as 2 box pull. DLNA 1.5 defines M-DMP, a DMP profile dedicated to mobile applications.

**Digital Media Printer DLNA 1.5 (DMPr)** is a device or program providing document and image printing services to the other devices.

**Digital Media Renderer UPnP AV, DLNA (DMR)** is a device or program rendering content it receives after being setup by another network entity. See below use case defined by DLNA as 2 box push. DLNA 1.5 defines M-DMR, a DMR profile dedicated to mobile applications.

**Digital Media Server UPnP AV, DLNA (DMS)** is a device or program exposing and distributing content throughout the home. DLNA 1.5 defines M-DMS, a DMS profile dedicated to mobile applications

**Digital Right Managements (DRM)** is any of several technologies used by publishers (or copyright owners) to control access to and usage of digital data (such as software, music, movies) on hardware, handling usage restrictions associated with a specific instance of a digital work. The term often is confused with copy protection and technical protection measures. These two terms refer to technologies that control or restrict the use and access of digital media content on electronic devices with such technologies installed, acting as components of a DRM design.

**Digital Transmission Content Protection over Internet Protocol (DTCP/IP)** is a specification for link protection of copyrighted content that is transferred over digital interfaces in home networks that adhere to IP. Under this specification, digital content can be shared securely between devices in a user's home but not shared with third-parties outside the home network. Using an authentication scheme, DTCP-IP allows the user to designate devices in the home network as trusted destinations that can stream data back and forth, but DTCP-IP will not allow the content to be transmitted outside of the home network or copied in any manner.

**Digital Video Broadcasting (DVB)** is a suite of internationally accepted, open standards for digital television maintained by the DVB Project, an industry consortium with more than 270 members, and published by a Joint Technical Committee (JTC) of European Telecommunications Standards Institute (ETSI), European Committee for Electrotechnical Standardization (CENELEC) and European Broadcasting Union (EBU).

**Digital Video Recorder (DVR)**, see PVR.

**EPG (Electronic Program Guide)** is an on-screen guide to scheduled broadcast television programs, allowing a viewer to navigate, select, and discover content by time, title, channel, genre, etc, by use of their remote control, or a keyboard.

**ESG (Electronic Service Guide)**, see EPG.

**GOD (Games on Demand)** is to computer games as video on demand is to video rental or video purchase.

**High-Bandwidth Digital Content Protection (HDCP)** is a link protection developed by Intel Corporation to protect digital audio and video content as it travels across wired baseband video connections such as Digital Visual Interface (DVI), High Definition Multimedia Interface (HDMI) and DisplayPort (DP) connections. The HDCP specification is proprietary and an implementation of HDCP requires a license.

**High-Definition Multimedia Interface (HDMI)** is an all-digital audio/video interface capable of transmitting uncompressed baseband A/V. HDMI provides an interface between any compatible digital audio/video source, such as a set-top box, a DVD player, a PC, a video game system such as the PlayStation 3 or an AV receiver and a compatible digital audio and/or video monitor, such as a digital television (DTV) or A/V Receiver (AVR).

**HyperText Transfer Protocol (HTTP)** is a request/response protocol between clients and servers. The originating client, such as a web browser, spider, or other end-user tool, is referred to as the user agent. The destination server, which stores or creates resources such as HTML files and images, is called the origin server. In between the user agent and origin server may be several intermediaries, such as proxies, gateways, and tunnels.

**Internet Radio** is a broadcasting service transmitted via the Internet. Not every internet "radio station" has a corresponding traditional radio station. Many internet radio stations are completely independent from traditional ("terrestrial") radio stations and broadcast only on the Internet. Broadcasting on the Internet is usually referred to as streaming.

**Interactive Program Guide (IPG)**, see EPG.

**Mobile Digital Media Uploader (M-DMU)** *DLNA 1.5* is a device DLNA class with mobile handheld environmental characteristics that allows to upload content to a Media server.

**Media Center** is a device adapted for playing music, watching movies and pictures stored on a local hard drive or on a network, watching DVD movies and often for watching and recording television broadcasts. Some software is capable of doing other tasks, such as finding news (RSS) from the internet. Media centers are operated with a remote control, connected to a television set for video output.

**Media Center Extender**, see DMA.

**(Media Interoperability Unit) M-MIU** From *DLNA 1.5*, is a deprecated device DLNA class that provides media format interoperability between mobile and stationary devices.

**Mobile Network Connectivity Function (M-NCF)** From *DLNA 1.5*, is a deprecated device DLNA class with mobile handheld environmental characteristics that provides interoperability by bridging the network connectivity layer between mobile and stationary devices.

**Music On Demand (MOD)** is a music distribution model conceived with the growth of two-way computing, telecommunications and the Internet in the early 1990's. Primarily, high-quality music is made available to purchase, access and playback using software on the Apple Macintosh, Microsoft Windows, set-top boxes and mobile devices from an available distribution point, such as a computer host or server located at a telecom, cable TV or wireless data center facility.

**Network Attached Storage (NAS)** is the name given to dedicated data storage technology that can be connected directly to a computer network to provide centralized data access and storage to

heterogeneous network clients. A NAS can belong to the class of Digital Media Server in DLNA vocabulary.

**Network Music Player** is a hardware device designed to play internet radio or digital audio streamed to it across a home network.

**Personal Video Recorder (PVR)** is a device that records video without videotape to a hard drive-based digital storage medium. The term includes stand-alone set-top boxes and software for personal computers which enables video capture and playback to and from disk.

**Quality of Service (QoS)** refers to the probability of the telecommunication network meeting a given traffic contract, or in many cases is used informally to refer to the probability of a packet succeeding in passing between two points in the network within its desired latency period.

**Reference Design** refers to a technical blueprint of a system that is intended for others to copy. It contains the essential elements of the system; however, third parties may enhance or modify the design as required.

**Remote UI UPnP** sends and receives control-commands between UPnP-client and UPnP-server over network (such as record, schedule, play, pause, stop, etc.).

**Residential Gateway** is a hardware device that connects a home or small office network to the Internet. The residential gateway provides port translation (NAT) and allows all the computers in a small network to share one IP address and Internet connection. The residential gateway may sit between the modem and the internal network, or a DSL or cable modem may be integrated into the residential gateway. A residential gateway often combines the functions of an IP router, multi-port Ethernet switch and Wi-Fi access point.

**RTP(Real Time Protocol)** is a network protocol for delivering audio and video. RTP does not guarantee real-time delivery of data, but provides mechanisms for the sending and receiving applications to support streaming data.

**Set-Top Box** a device that connects to a television and some external source of signal, and turns the signal into content then displayed on the TV screen. The signal source might be an Ethernet cable, a satellite dish, a coaxial cable, a telephone line, or even an ordinary VHF or UHF antenna. Content, in this context, could mean any or all of video, audio, Internet web pages, interactive games, or other possibilities. A set-top box does not necessarily contain a tuner of its own, but can.

**Streaming Media** is media that is heard or viewed while it is being delivered. Streaming is more a property of the delivery system than the medium itself. The distinction is usually applied to media that are distributed over computer networks and are rendered with little delay from being delivered; file transfer is a form of streaming during the copy process. Once the file is copied locally it is no longer considered as a stream. Most other delivery systems are either inherently streaming (radio, television) or inherently non-streaming (books, video cassettes, audio CDs).

**UPnP (Universal Plug and Play)** is an architecture for pervasive peer-to-peer network connectivity of intelligent devices, particularly within the home. UPnP performs device and service discovery and control through driverless, standards-based protocol mechanisms. Universal Plug and Play devices can automatically configure network addressing, announce their presence on a network subnet, and permit the exchange of device and service descriptions. UPnP defines three classes of devices: Audio/Video (AV), Internet Gateway Devices (IGD), Printers/Scanners, & Home Automation.

**UPnP AV** *UPnP* is an architecture for pervasive peer-to-peer network connectivity of intelligent devices, particularly within the home. UPnP performs device and service discovery and control through driverless, standards-based protocol mechanisms. Universal Plug and Play devices can automatically configure network addressing, announce their presence on a network subnet, and permit the exchange of device and service descriptions. The goals of UPnP are to allow devices to connect seamlessly and to simplify the implementation of networks in the home (data sharing, communications, and entertainment) and corporate environments. UPnP achieves this by defining and publishing UPnP Device Control Protocols built upon open, Internet-based communication standards. UPnP defines three classes of devices: Audio/Video (AV), Internet Gateway Devices (IGD), Printers/Scanners, & Home Automation.

**Video On Demand (VOD)** allows users to select and watch video content over a network as part of an interactive television system. VOD systems either "stream" content, allowing viewing while the video is being transmitted, or "download" it in which the program is brought in its entirety to a set-top box before viewing starts.

**Windows Media Digital Right Management 10 (WMDRM)** is Microsoft implementation of digital right management. It covers two different uses: Portable Devices (PD, alias Janus) and Network Devices (ND, alias Cardea). PD is a digital right management technology strictly speaking, whereas ND is a link protection technology.

**Wireless Multimedia Extensions (WMM)** Within the Wi-Fi Alliance is an interoperability certification, based on the IEEE 802.11e draft standard. It provides basic Quality of service (QoS) features to IEEE 802.11 networks. WMM prioritizes traffic according to 4 AC (Access Categories) - voice, video, best effort, and background. However, it does not provide guaranteed throughput.

*Entries in this glossary licensed under the GNU Free Documentation License. Material from Wikipedia articles is used.*

