

ETSI TS 103 286-3 V1.1.1 (2015-05)



**Digital Video Broadcasting (DVB);  
Companion Screens and Streams;  
Part 3: Discovery**

**EBU**  
OPERATING EUROVISION

**DVB**<sup>®</sup>  
Digital Video  
Broadcasting

---

**Reference**DTS/JTC-DVB-343-3

---

**Keywords**broadcast, companion screen, synchronization,  
TV**ETSI**650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

---

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

---

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

---

**Copyright Notification**

---

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2015.

© European Broadcasting Union 2015.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

---

# Contents

Intellectual Property Rights .....	4
Foreword.....	4
Modal verbs terminology.....	4
Introduction .....	5
1 Scope .....	6
2 References .....	6
2.1 Normative references .....	6
2.2 Informative references.....	7
3 Definitions and abbreviations.....	7
3.1 Definitions .....	7
3.2 Abbreviations .....	8
4 Overview .....	9
4.1 Introduction .....	9
4.2 UPnP™ Device Architecture.....	9
4.3 UPnP™ Application Management .....	10
4.4 CSS-CII interface mapping on Application Management .....	11
4.5 CSA behaviour .....	12
5 UPnP™ Device requirements for the TV Device .....	14
History .....	15

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECTrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

**NOTE:** The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union  
CH-1218 GRAND SACONNEX (Geneva)  
Switzerland  
Tel: +41 22 717 21 11  
Fax: +41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprising over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

The present document is part 3 of a multi-part deliverable covering the DVB Companion Screens and Streams Specification, as identified below:

- Part 1: "Concepts, roles and overall architecture";
- Part 2: "Content Identification and Media Synchronization";
- Part 3: "Discovery".**

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

## Introduction

It is assumed that the reader is familiar with parts 1 and 2 of this multi-part deliverable that provide background information on the concepts covered in the present document.

Personal, smart devices like tablet computers and smart phones enable new user experiences for broadcast service consumption. Many of these new experiences require communication via an IP network between applications or services on a TV Device and Companion Screen Applications (CSAs) on a personal device.

An application or service typically needs to know the network location to direct communications to if it is to reach the intended application or service on the intended device. A location is, at minimum, described by the assigned network IP address and may also include other details such as an IP port number or a resource identifying path. On a home IP network, the devices are typically assigned network addresses by means of static addressing or dynamic allocation via the Dynamic Host Configuration Protocol (DHCP). Hence, it is not known in advance which addresses should be used for communication. It is also not desirable to expect the user to have knowledge of network addresses and to input these into the device.

The present document describes the technology used to establish communication between the TV Device and CSA on a companion device without prior knowledge of the assignment of network addresses. This enables a CSA to establish communication with the TV Device via interfaces defined in other parts of this specifications series and therefore to provide the experiences that those interfaces enable.

The technology used to achieve this is Universal Plug and Play (UPnP™) and it is assumed that the reader is familiar with architecture and technologies used in UPnP™ ([1] and [2]).

---

# 1 Scope

The present document specifies the architecture and protocols for discovering TV Devices in the home that implement, interfaces defined in the other parts of this multi-part deliverable. The discovery protocol is used by the Companion Screen Application to detect TV Devices.

The discovery protocol supplies the information so that the following interfaces can be used over the established link:

- the interfaces between the TV Device and the Companion Screen Application:
  - interface for metadata exchange, including content identification (CSS-CII);
  - interface for Wall Clock synchronization (CSS-WC);
  - interface for Timeline Synchronization (CSS-TS);
  - interface for notification of Trigger Events (CSS\_TE).

The interfaces themselves are described in clauses 6, 8, 9 and 10 of ETSI TS 103 286-2 [4].

Note that Wall Clock synchronization and Timeline Synchronization interfaces are (currently) discovered indirectly. The present document defines the means to discover the CSS-CII interface. The other interfaces are then discovered by means of the established connection for the CSS-CII interface and this is defined in ETSI TS 103 286-2 [4].

The present document does not describe:

- The UPnP™ Device Architecture [1].
- The UPnP™ Application Management service [2].
- How to establish communication through a web socket [3].

---

## 2 References

### 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

- [1] ISO/IEC 29341-1:2011: "Information technology -- UPnP Device Architecture -- Part 1: UPnP Device Architecture Version 1.0".

NOTE: This specification was first published by UPnP™ in 2008, and an equivalent version is available from <http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf>.

- [2] UPnP™ Forum (September 2014): "ApplicationManagement:1 Service".

NOTE: Available at: <http://upnp.org/specs/ms/UPnP-ms-ApplicationManagement-v1-Service.pdf>. This specification is expected to be published by ISO/IEC, though this has not happened at the time of publication of the present document.

- [3] IETF RFC 6455 (December 2011): "The WebSocket Protocol".

NOTE: Available at <http://tools.ietf.org/html/rfc6455>.

- [4] ETSI TS 103 286-2: "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 2: Content Identification and Media Synchronization".

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI TS 103 286-1: "Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 1: Concepts, roles and overall architecture".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

**action:** named subroutine

NOTE: This is UPnP™ terminology. The action is based on the RPC mechanism defined in the UDA.

**AppInfoList:** XML document which describes an application

NOTE: This is UPnP™ terminology specific to the Application Management DCP.

**application:** software program designed to perform an activity

NOTE: This is UPnP™ terminology.

**Application Management DCP:** UPnP™ DCP that allows applications to be installed, started and stopped by UPnP™ Actions

NOTE: This is UPnP™ terminology.

**app-to-app communication:** communication between two applications not using the UPnP™ Device Architecture

NOTE: This is terminology from the specification of the UPnP™ Application Management service definition for a protocol that is out of band with respect to the UPnP™ Device Architecture.

**connectionAddress:** connection end point of the protocol

NOTE: This is UPnP™ terminology. It is the name of an XML property in the AppInfoList.

**Control Point:** UPnP™ client

NOTE: This is UPnP™ terminology. A Control Point represents the client side of the client server solution defined by the UPnP™ Device Architecture.

**ControlURL:** HTTP end point for SOAP actions for the UPnP™ Service

NOTE: This is UPnP™ terminology. The controlURL is conveyed in a DDD.

**event:** notification of a state change from a UPnP™ Device to a Control Point

NOTE: This is UPnP™ terminology.

**M-Search:** mechanism to search for a UPnP™ Device or UPnP™ Service

NOTE: This is UPnP™ terminology. M-Search is part of SSDP discovery protocol.

**matchingProtocolName:** identification of the supported protocol over the connection

NOTE: This is UPnP™ terminology. It is the name of an XML property in the AppInfoList.

**runningStatus:** identification of the status of the application

NOTE: This is UPnP™ terminology. It is the name of an XML property in the AppInfoList.

**response:** UPnP™ Device created result including output parameters as result of an action invocation

**request:** invocation of an action by Control Point describing which action is being called on the UPnP™ Device including the input parameters

**UPnP™ Device:** logical UPnP™ entity on the network that exposes functionality through UPnP™ Services.

NOTE: This is UPnP™ terminology. A UPnP™ Device can uniquely be identified on the network by means of an UDN.

**UPnP™ Device Type:** definition of the type of a UPnP™ Device

NOTE: This is UPnP™ terminology. UPnP™ Device Types can mandate implementation of specific UPnP™ Services.

**UPnP™ Service:** logical collection of UPnP™ Actions and State variables that makes up functionality in a domain

NOTE: This is UPnP™ terminology.

## 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CSA	Companion Screen Application
CSS	Companion Screens and Streams
CSS-CII	Interface for Content Identification and other Information
DCP	Device Control Protocol
DHCP	Dynamic Host Configuration Protocol
DDD	Device Description Document
GENA	General Event Notification Architecture
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MPEG	Multimedia Pictures Expert Group
RPC	Remote Procedure Call

NOTE: UPnP™ uses SOAP as RPC mechanism.

SCPD	Service Control Protocol Description
SOAP	Simple Object Access Protocol.

NOTE: SOAP is the definition of how to model an RPC subroutine in XML.

SSDP	Simple Service Discovery Protocol
TCP	Transmission Control Protocol
UDA	UPnP™ Device Architecture
UDP	User Datagram Protocol
UDN	Unique Device Name

NOTE: The UDN is carried as a property that is always included in the DDD.

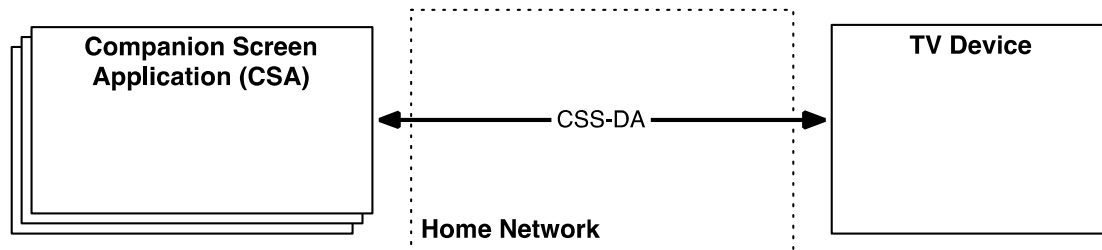
UPnP™	Universal Plug and Play
XML	eXtensible Markup Language



## 4 Overview

### 4.1 Introduction

Universal Plug and Play (UPnP™) is a well known and established protocol to find and identify UPnP™-enabled devices on the network. UPnP™ is used to convey the information to enable the CSA to discover and associate with the TV Device as depicted in Figure 1, and thereby establish communication. See also ETSI TS 103 286-1 [i.1].



**Figure 1: CSA and TV Device connected to home network.**

The following clauses provide an informative descriptive overview of the discovery functionality in the light of the device requirements defined in clause 5.

Clause 4.2 provides a brief overview of the main elements of the UPnP™ Device Architecture.

Clause 4.3 describes the use and operation of the UPnP™ Application Management as profiled in the present document and implemented of by the TV Device.

Clause 4.4 explains how the location of the endpoint for the CSS-CII interface (defined in ETSI TS 103 286-2 [4]) is exposed through UPnP™ Application Management.

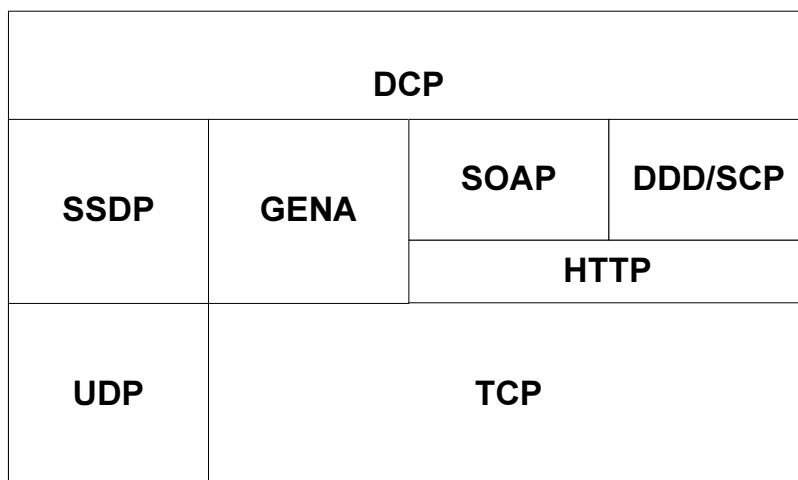
Clause 4.5 illustrates the sequence of operations that a CSA can use to discover and associate with the TV Device and thereby determine the location of the CSS-CII endpoint.

### 4.2 UPnP™ Device Architecture

UPnP™ implements a client server model. In UPnP™ terminology, clients are called Control Points and servers are called UPnP™ Devices. The functionality of UPnP™ Devices and Control Points is abstracted from the transport mechanism. The functionality is described in Device Control Protocols (DCPs) that define a collection of Actions (RPC methods) and Events. The transport mechanism is described in the UPnP™ Device Architecture (UDA) [1] and the components that make up the UDA are illustrated in Figure 2. The UDA describes:

- how Control Points can detect UPnP™ Devices on the network without prior knowledge of the device IP address using the Simple Service Discovery Protocol (SSDP) that takes place via UDP;
- which capabilities the detected UPnP™ Devices implement (by means of downloading an XML Device Description Document and XML Service Control Protocol Description document by means of HTTP);
- how Control Points interact with the exposed functionality by invoking actions using SOAP requests via HTTP; and
- how Control Points receive events using GENA via TCP.

The domain specific functions are described per domain in DCPs. The DCPs are specified as UPnP™ Services in a UPnP™ Device.



**Figure 2: Schematic overview of a UPnP™ Stack**

### 4.3 UPnP™ Application Management

The UPnP™ Application Management service DCP [2] is used to establish the communication between the CSA and the TV Device. The TV Device implements a UPnP™ Device that incorporates the Application Management service and the CSA implements an Application Management Control Point. This is illustrated in Figure 3.

**NOTE:** Application Management is a UPnP™ term and refers to a service for managing abstract entities known as applications. An application is defined by individual profiles and implementations of the Application Management service DCP. Therefore it should not be confused with the management of applications that run on a companion device.

What follows is a brief overview of the Application Management service. How this service is used to expose the location of the CSS-CII interface endpoint is described in clause 4.3.

The UPnP™ Application Management service specification describes functionality to establish App-to-App communication. The Application Management service has mechanisms to:

- determine which applications are supported by the implementation;
- determine which applications are currently running;
- retrieve detailed information about an application, including which kind of app-to-app protocols are supported by each application;
- retrieve the information needed to establish the Out-of-Bound communication for a given app-to-app protocol.

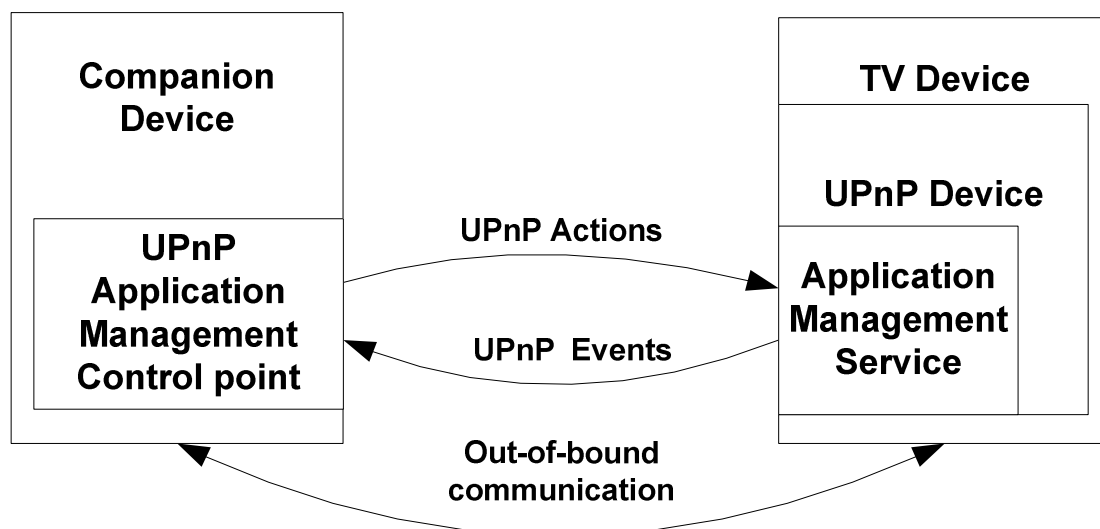


Figure 3: Typical implementation of an Application Management Control Point in a Companion Device and the UPnP™ Application Management service in a TV Device

#### 4.4 CSS-CII interface mapping on Application Management

The CSA acts as an Application Management Control Point. This will allow the CSA to detect the TV Device that will implement the UPnP™ Device that implements the Application Management service DCP. This is depicted in Figure 4.

The Application Management service in the UPnP™ Device within the physical TV Device will list a DVB CSS-CII application. This application will be identified by means of the matchingProtocolName.

The CSS-CII interface is discovered by finding the application with the correct matchingProtocolName in any Application Management service. The CSS-CII interface is WebSocket based; other CSS interfaces may use different transport mechanisms than a WebSocket.

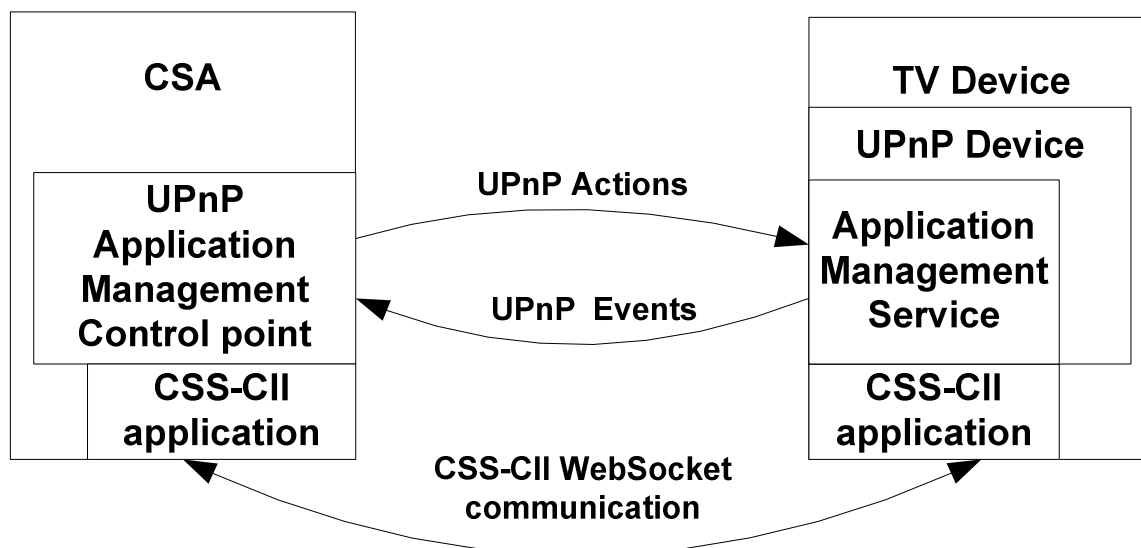


Figure 4: CSA and TV Device implementing UPnP™ Application Management and CSS-CII application communication

## 4.5 CSA behaviour

The determination of the CSS-CII WebSocket address from a CSA towards a TV Device is achieved by the following sequence of steps as depicted in Figure 5:

1. Finding the TV Device on the home network, by issuing an M-Search.
2. Selecting a TV Device by filtering the responses of the M-Search.
3. Downloading the DDD of the selected TV Device.
4. Issuing Actions to determine:
  - a. If an application exist with matchingProtocolName by invoking Action GetAppIDList().
  - b. Retrieve the application information that contains the WebSocket address and the runningState by invoking action GetAppInfoByIDs().
5. Establish the CSS-CII WebSocket connection.

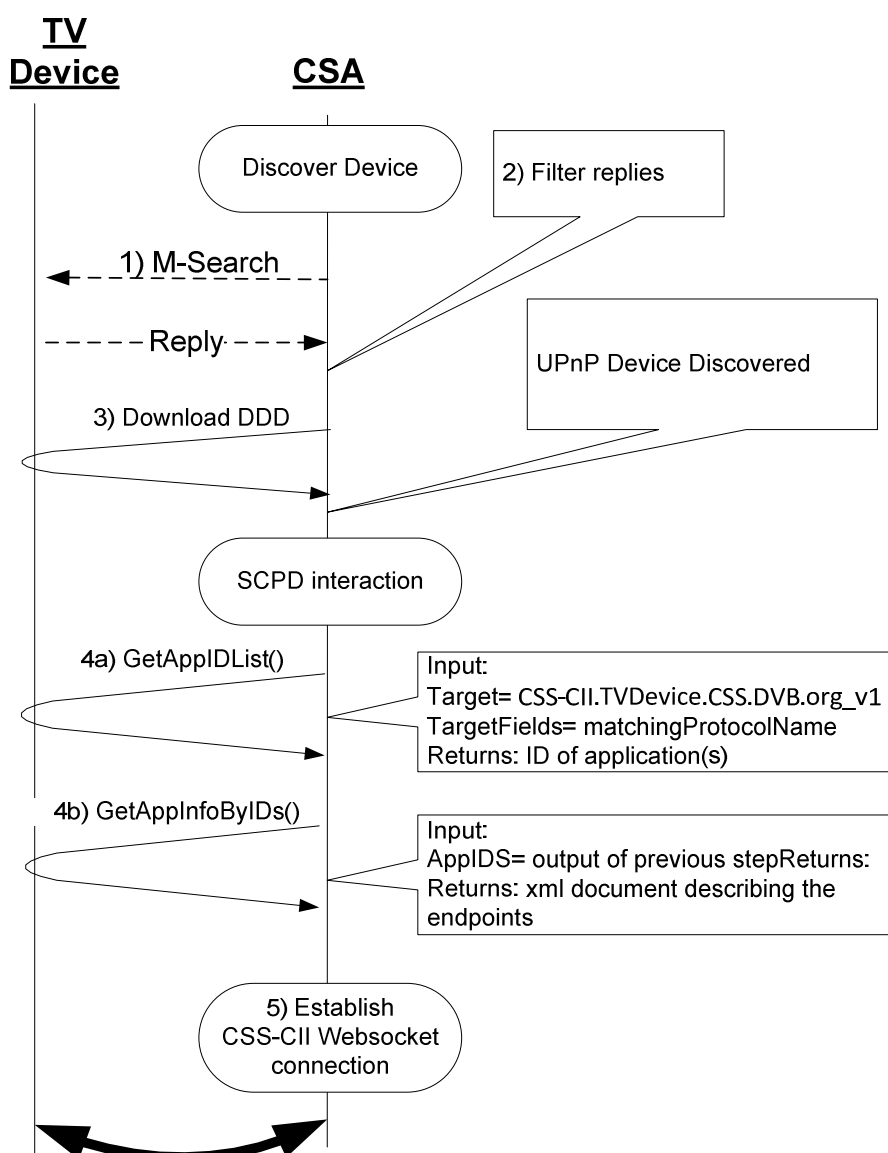


Figure 5: Sequence of commands between CSA and TV Device to establish a WebSocket connection

It is not known whether the CSA or the TV Device started first on the network. Therefore, typically, the CSA implements two mechanisms to detect the TV Device:

- 1) The CSA can monitor ssdp:alive messages for the Application Management service. This allows the TV Device to be detected by the CSA when the TV Device is entering the network and the CSA has already entered the network.
- 2) The CSA can search for the Application Management service when the CSA enters the network and find already started TV Devices.

The Application Management service implemented by any UPnP™ Device type can be found by issuing an M-Search with one of the search targets defined as one of the following:

- "ssdp-all", this will include also the replies of each implemented service.
- "urn:schemas-upnp-org:service:ApplicationManagement:1", this will result in only in replies of UPnP™ Devices that implement this particular service. Note that higher versions of the service will be found as well.

The Control Point can filter the SSDP responses on the Application Management service identifier to find the UPnP™ Device that contains the Application Management service.

The Control Point can find a previously found TV Device on the network by issuing an M-Search with the UDN as search target. The UDN is conveyed in the DDD and remains the same over reboots of a UPnP™ Device. To associate a previous selected TV Device the UDN of the TV Device needs to be stored persistently by the CSA. This will enable a CSA to provide its user with the facility to automatically reconnect to, for example, the TV Device they selected the previous time they used the CSA.

The Control Point uses the value of the location header of the SSDP reply or alive message to download the DDD. The DDD contains the controlURL for each implemented service. The Control Point may use the controlURL of the Application Management service to invoke actions on the Application Management service implemented in the UPnP™ Device.

The Control Point uses the Actions of the Application Management service to find the application identified by matchingProtocolName "CSS-CII.TVDevice.CSS.DVB.org\_v1" and verify that the found application has the runningState "Running". These preconditions need to be met before establishing the WebSocket connection for the CSS-CII interfaces as described in clause 6 of ETSI TS 103 286-2 [4] by using the WebSocket address supplied by the application's connectionAddress.

In the following example the CSA Control Point uses the GetAppIDList() action to determine if the Application Management service has implemented version 1 of the CSS-CII TV Device application as defined as: "CSS-CII.TVDevice.CSS.DVB.org\_v1" and has application id "110E4EC1-6CC4-4D12-9995-7F996B709726".

#### Request:

```
GetAppIDList (Target="CSS-CII.TVDevice.CSS.DVB.org_v1", TargetFields="matchingProtocolName")
```

#### Response:

```
GetAppIDList (AppIDs = "110E4EC1-6CC4-4D12-9995-7F996B709726")
```

The returned AppIDs are used as input for the GetAppInfoByIDs() action to retrieve the XML description of each returned application id. In this case only 1 application id is returned.

#### Request:

```
GetAppInfoByIDs (Target="110E4EC1-6CC4-4D12-9995-7F996B709726")
```

#### Response:

```
GetAppInfoByIDs (AppInfo =
  "<?xml version="1.0" encoding="UTF-8"?>
  <AppInfoList
    xmlns="urn:schemas-upnp-org:ms:ams" xsi:schemaLocation="
    urn:schemas-upnp-org:ms:ams">
  <application id="110E4EC1-6CC4-4D12-9995-7F996B709726">
    <friendlyName>TV Device CSS-CII DVB Application</friendlyName>
    <runningStatus>Running</runningStatus>
  </application>
  </AppInfoList>")
```

```

    <matchingProtocolName>CSS-CII.TVDevice.CSS.DVB.org_v1</matchingProtocolName>
    <protocol required="1">Websocket</protocol>
    <connectionAddress>ws://192.168.0.1:80/MyCSSCIIDVBApp</connectionAddress>
  </apptoAppInfo>
</application>
</AppInfoList>")

```

The returned XML document contains the found TV Device CSS-CII application in the running state and contains a web socket address `ws://192.168.0.1:80/MyCSSCIIDVBApp`.

NOTE: For readability the example XML above is not XML escaped.

---

## 5 UPnP™ Device requirements for the TV Device

The TV Device shall implement a UPnP™ Device that implements the Application Management service DCP. The implemented UPnP™ Device shall comply with UPnP™ Device Architecture [1] or a later version of the UDA.

NOTE 1: The Application Management service can reside in any UPnP™ Device Type. It is up to the vendor of the UPnP™ Device to choose an appropriate UPnP™ Device type.

The Application Management service shall comply with the service described in [2] or a later version. This means that the mandatory actions and state variables in the Application Management service shall be implemented.

The CSS-CII interface shall be presented as an application in the Application Management service.

The following requirements shall apply for this application:

- The application shall provide an <apptoAppInfo> element within XML documents describing the application where:
  - the matchingProtocolName shall be defined as "CSS-CII.TVDevice.CSS.DVB.org\_v1"
  - the protocol shall be defined as "Websocket"
  - the value of the protocol required attribute shall be defined as "1"
  - the connectionAddress shall be a valid WebSocket address
- The connectionAddress shall point at the WebSocket end point as defined in section 6.3 of ETSI TS 103 286-2 [4] and shall convey the CSS-CII protocol as defined in section 6.2 of ETSI TS 103 286-2 [4].
- The application shall always have the runningState of "Running".
- When the implementation implements the action StopApp(), the application shall not be stopped when the action StopApp() is invoked. This function shall return error code 710.

NOTE 2: It is not required to implement StopApp().

- The XML document describing the application as result of invoking the action GetAppInfoByIDs() shall always include the properties:
  - runningStatus
  - apptoAppInfo

---

## History

<b>Document history</b>		
V1.1.1	May 2015	Publication