ETSI TS 103 544-12 V1.3.1 (2019-10)

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 12: UPnP Server Device

CAUTION

The present document has been submitted to ETSI as a PAS produced by CCC and approved by the ETSI Technical Committee Intelligent Transport Systems (ITS).

Reference
RTS/ITS-98-12

Keywords
interface, ITS, PAS, smartphone

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 prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

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

Contents

Intellectual Property Rights

Essential patents

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 (https://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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 12 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.1].

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.

1 Scope

The present document is part of the MirrorLink® specification which specifies an interface for enabling remote user interaction of a mobile device via another device. The present document is written having a vehicle head-unit to interact with the mobile device in mind, but it will similarly apply for other devices, which provide a color display, audio input/output and user input mechanisms.

The present document defines the device:

urn:schemas-upnp-org:device:*TmServerDevice:1*.

This device can be a UPnP root device or embedded within a different device.

The *TmServerDevice* encapsulates all services for the MirrorLink UPnP Server Device Control Protocol (DCP).

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 referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://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] UPnP™ Forum: "UPnP™ Device Architecture 1.1", 15 October 2008.

NOTE: Available at http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

[2] ETSI TS 103 544-26 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 26: Consumer Experience Principles and Basic Features".

[3] W3C Recommendation 11 April 2013: "XML Signature Syntax and Processing Version 1.1".

NOTE: Available at http://www.w3.org/TR/xmldsig-core/.

[4] Unicode Consortium: "Unicode 12.1 Character Code Charts".

NOTE: Available at http://www.unicode.org/charts/.

[5] ETSI TS 103 544-4 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 4: Device Attestation Protocol (DAP)".

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 referenced 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 103 544-1 (V1.3.1): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

Void.

4 Device Definitions

4.1 Device Type

The following device type identifies a device that is compliant with this template:

urn:schemas-upnp-org:device:*TmServerDevice:1*

We herein refer to this device in the present document as *TmServerDevice*. The *TmServerDevice* device shall follow defined UPnP behaviour within the UPnP Device Architecture 1.1 [1].

4.2 Device Model

4.2.1 General

Table 1 briefly describes the services used in *TmServerDevice*.

Table 1: TmServerDevice Service Descriptions

Service Name	Service Description
TmApplicationServer	Allows for discovery and remote control of applications.
TmClientProfile	Allows MirrorLink UPnP Control Point to specify its preferences, settings and capabilities.

Service Name	Service Description
TmNotificationServer	Allows MirrorLink UPnP Server to send notification events.

Products that expose devices of the type **urn:schemas-upnp-org:device:***TmServerDevice*:1 shall implement minimum version numbers of all required embedded devices and services specified in Table 2.

Table 2: Device Requirements for TmServerDevice

DeviceType	Root	Req. or Opt. (note 1)	ServiceType	Req. or Opt. (note 1)	Service ID (note 2)
TmServerDevice:1	Yes	R	TmApplicationServer:1	R	TmApplicationServer
			TmClientProfile:1	R	TmClientProfile
			TmNotificationServer:1	O	TmNotificationServer
NOTE 1: R = Required, O = Optional. NOTE 2: Prefixed by urn:upnp-org:service:.					

4.2.2 Relationship Between Services

Figure 1 shows the logical structure of the device and the encapsulated services which provide MirrorLink capabilities.

The *TmClientProfile* service provides a way for the MirrorLink UPnP Control Point to notify the MirrorLink Server device about its preferences, capabilities and desired settings (i.e. client profile). This information can then be utilized by other services hosted by the MirrorLink Server device such as the *TmApplicationServer* service.

The *TmApplicationServer* service provides a way for the MirrorLink UPnP Control Point to remotely control and access applications on the MirrorLink Server device.

The *TmNotificationServer* service provides a way for the MirrorLink UPnP Server to notify the MirrorLink UPnP Control Point on application notification events.

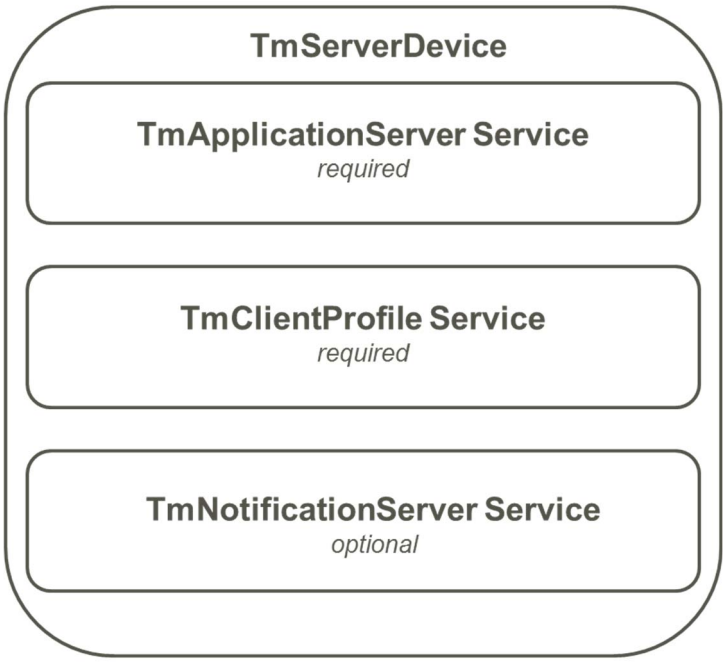

Figure 1: Relationship between *TmServerDevice* and Services

4.3 Theory of Operation

4.3.1 General

TmServerDevice provides mechanisms which enable MirrorLink UPnP Control Points to discover and access MirrorLink services.

Table 3 lists the attributes which are part of the *TmServerDevice* and specified as extensions to the standard UPnP Device XML schema.

Table 3: Extended Attributes for TmServerDevice

Element	Description	Parent	Availability
X_mirrorLink Version	MirrorLink Server Version Note: If the version information is missing, the MirrorLink Client shall assume a version 1.0 MirrorLink Server.	device	Mandatory
majorVersion	MirrorLink Server Major Version A_ARG_TYPE_Int	X_mirrorLink Version	Mandatory
minorVersion	MirrorLink Server Minor Version A_ARG_TYPE_Int	X_mirrorLink Version	Mandatory
X_connectivity	Connectivity settings	device	Conditional
bluetooth	Bluetooth settings of device	X_connectivity	Conditional
bdAddr	Bluetooth MAC address (BD_ADDR). Indicates device support for Bluetooth on the device. (A UTF-8 encoded string representing an unsigned 48-bit integer in hexadecimal format (without any '0x' prefix).)	bluetooth	Conditional
startConnection	A_ARG_TYPE_Bool Bluetooth Connection will be initiated from device. **Default**: `true`	bluetooth	Optional
clientBdAddr	Bluetooth MAC address (BD_ADDR) of the connected Bluetooth device. (A UTF-8 encoded string representing an unsigned 48-bit integer in hexadecimal format (without any '0x' prefix).) Shall be included, if *bdAddr* is not provided and a Bluetooth connection exists.	bluetooth	Conditional
wifi	WiFi settings of the device	X_connectivity	Optional
macAddr	WiFi MAC address (A UTF-8 encoded string representing an unsigned 48-bit integer in hexadecimal format (without any `"0x"` prefix, and without any grouping using `":"`, `"."` or `"-"`)	wifi	Mandatory
ssid	Service Set Identifier (SSID), Base64 encoded (A_ARG_TYPE_String)	wifi	Optional
roles	Comma separated list of supported roles. Allowed values are • `AP` (Access Point role) • `Client` (Client role) • `P2P` (Infrastructure-less) (A_ARG_TYPE_String) **Default**: `AP,Client,P2P`	wifi	Optional
protectionList	List of WiFi access protection	wifi	Optional

Element	Description	Parent	Availability
protection*	Access protection	protectionList	Optional
protocol	Security protocol used to protect WiFi access. Allowed values are • WEP • WPA • WPA2 • WPS NOTE: WEP/WPA is listed for legacy reasons, and should not be used (A_ARG_TYPE_String)	protection	Mandatory
passkey	Passkey/Shared key, Base64 encoded Shall be left empty, if transmitted over an unprotected or shared transport channel (e.g. WiFi) (A_ARG_TYPE_String)	protection	Mandatory
X_deviceKeys	Device specific physical hard keys **Deprecated**	device	Deprecated
key*	Defines a device specific key **Deprecated**	X_deviceKeys	Deprecated
name	Short name (A_ARG_TYPE_String) **Deprecated**	key	Deprecated
mandatory	Flag (A_ARG_TYPE_Bool) **Deprecated**	key	Deprecated
symbolValue	Key's symbol hexadecimal value **Deprecated**	key	Deprecated
icon*	Describes an icon representing the key **Deprecated**	key	Deprecated
mimetype	Type of icon image **Deprecated**	icon	Deprecated
width	Width of icon (A_ARG_TYPE_INT) **Deprecated**	icon	Deprecated
height	Height of icon (A_ARG_TYPE_INT) **Deprecated**	icon	Deprecated
depth	Icon color depth (A_ARG_TYPE_INT) **Deprecated**	icon	Deprecated
url	Url to icon (A_ARG_TYPE_URI) **Deprecated**	icon	Deprecated
X_Signature	XML signature over entire contents of the *root* element. This is done as specified in [3]. The key used in calculating the signature shall be the private part of the application-specific key which public part was bound to the attestation of UPnP-Server component. (The public part can be used to verify the signature.) The Reference element of the XML signature shall be empty. The *SignatureMethod* shall be RSA with SHA1. The *KeyInfo* element may be omitted. The mechanism for generation, exchange and maintenance of keys is out of scope for the present document.	device	Mandatory
X_presentations	Presentation protocols supported from the MirrorLink Server. MirrorLink Server shall include this element if it supports presentation protocols other than "vncu". (MirrorLink 1.2)	device	Mandatory

Element	Description	Parent	Availability
presentation	Comma-separated list of presentation protocols supported from the MirrorLink Server. • `hsml` • `wfd` • `vncu` • `vncw` (A_ARG_TYPE_String)	X_presentations	Mandatory
X_localization	Provide information about the localization support from the MirrorLink Server.	device	Optional
characterSet	Comma-separated list of entry points into the UniCode Character Code Charts, which are supported from the MirrorLink Server device. (UTF-8 encoded string; each entry point is given in hexadecimal format (with `"0x"` prefix).	X_localization	Mandatory
X_mlUiMode	Supported MirrorLink modes from the MirrorLink Server. Introduced in MirrorLink 1.3.	device	Mandatory
mode*	Supported MirrorLink mode from the MirrorLink Server. Allowed values are • `immersive` • `classic` (A_ARG_TYPE_String)	mlUiMode	Mandatory

The elements marked with an (*) can have multiple instances.

For deprecated values, the MirrorLink Server shall not include them into the UPnP Device XML. The MirrorLink Client shall ignore any content provided in deprecated elements.

The *modelNumber* element within the Device XML is a unique number identifying a family of devices, which expose identical MirrorLink related behavior, from the device manufacturer given in the *manufacturer* element. The model number format is vendor specific. It shall be smaller than 32 bytes. The *modelNumber* values are recorded by the CCC Certification Body.

Implementation Note

Some older MirrorLink Server devices need not provide a (unique) model number or a *manufacturer* element.

The MirrorLink Client shall validate the received *X_Signature.* A failure to successfully validate the *X_Signature* shall terminate the MirrorLink session.

NOTE: The public key needed to validate the received *X_Signature* is provided through the Device Attestation Protocol, bound to the *TerminalMode:UPnP-Server* component [5]. Therefore, the MirrorLink Client will either store (parts of) the received Device XML or retrieve it again.

If the MirrorLink Server has a Bluetooth module, the MirrorLink Server shall provide a Bluetooth MAC address (*bdAddr*), even if that module is not used within a potential MirrorLink connection.

Implementation Note

In case the underlying platform prevents access to the Bluetooth MAC address, as defined in the platform specific specification, the MirrorLink Server will need to handle some functionality on behalf of the MirrorLink Client. In this case, the MirrorLink Server shall disconnect BT A2DP, as soon as the MirrorLink Client is establishing an RTP forward or RTSP session.

A MirrorLink Server failing to include its Bluetooth MAC address may lead to the MirrorLink Client connecting to the wrong device, which is not the connected MirrorLink Server device. This may need to be resolved from the user, manually reconnecting to the correct device.

In case the MirrorLink Client only uses the MirrorLink Server's Bluetooth MAC address to determine, whether to use the MirrorLink Server's or the MirrorLink Client' s local in-Call UI, in case a phone call is established via Bluetooth HFP, the MirrorLink Client should default to its local in-Call UI.

The MirrorLink Server shall not use any default, wrong or non-existing Bluetooth MAC address, like `"02:00:00:00:00:00"`. This would otherwise lead the MirrorLink Client to a failing connection attempt.

XSD format descriptions shall be as given in Annex A.

The following applies to MirrorLink 1.2 and beyond.

The MirrorLink Server should provide information about its localization support with respect to the support of foreign language character sets. In case the information is provided, the MirrorLink Server shall include all supported character sets, as defined by the UniCode Character Code Chart given by the provided entry point, specified in [4].

NOTE: The Unicode code charts define a range for the respective code. The entry point is defined as the first value within that given range. E.g. Basic Latin (ASCII) has a range of 0x0000 - 0x007F. Therefore, its entry point is 0x0000.

A MirrorLink Server shall support all characters from a listed Code Chart. A MirrorLink Server shall support at least Basic Latin (ASCII), which is defined by the Character Code Chart entry `0x0000`.

The following applies from MirrorLink 1.3 onwards.

The MirrorLink Server shall provide the Bluetooth MAC address (*bdAddr*) of the connected Bluetooth device within the *clientBdAddr* element. In case that address is not accessible, the element shall be available, but left empty, e.g. `"</clientBdAddr>"`. In case the MirrorLink Server is not connected via Bluetooth, the element shall be omitted. The MirrorLink Client should use this information to detect the device, it is connected to via Bluetooth, and whether to display its local in-Call UI.

Table 4 shows the use of the *bdAddr* and *clientBdAddr* elements, dependent of the MirrorLink Server status. Status includes state of the Bluetooth connection as well as ability of the MirrorLink Server implementation to access the respective information.

Table 4: Announcement of MirrorLink Server and Client Bluetooth MAC Addresses

MirrorLink Server Status	*bluetooth*	*bdAddr*	*clientBdAddr*
No Bluetooth radio.	Shall be omitted.	N/A	N/A
Bluetooth radio. Access to *bdAddr*. Bluetooth not connected.	Shall be included.	Shall be Included. Shall be Valid, non-empty MAC address.	Shall be omitted.
Bluetooth radio. Access to *bdAddr*. Bluetooth connected. Access to *clientBdAddr*.			May be included. If included, shall be valid, non-empty MAC address.
Bluetooth radio. Access to *bdAddr*. Bluetooth connected. No access to *clientBdAddr*.			May be included. If included, shall be empty, e.g. `</clientBdAddr>`.
Bluetooth radio. No access to *bdAddr*. Bluetooth not connected.	Shall be included.	Shall be included. Shall be empty, e.g. `</bdAddr>`.	Shall be omitted.
Bluetooth radio. No access to *bdAddr*. Bluetooth connected. Access to *clientBdAddr*.			Shall be included. Shall be valid, non-empty MAC address.
Bluetooth radio. No access to *bdAddr*. Bluetooth connected. No access to *clientBdAddr*.			Shall be included. Shall be empty, e.g. `</clientBdAddr>`.

The Device XML is expected to be static. Therefore, the *clientBdAddr* shall represent the information at the time of the initial *SSDP:alive* advertisements.

The MirrorLink Server shall provide information about its supported MirrorLink Modes, as defined in [2]. The MirrorLink Server shall include all supported modes:

- `"immersive"` - support for Immersive MirrorLink Mode.
- `"classic"` - support for Classic MirrorLink Mode.

MirrorLink 1.1 and 1.2 Servers will not provide any MirrorLink Mode information. Those devices implement Legacy MirrorLink Mode. MirrorLink 1.3 Server devices shall support Classic MirrorLink Mode. Immersive MirrorLink Mode should be supported from the MirrorLink Server. More details are defined in [2].

4.3.2 XML Signature Minimum Set

The MirrorLink Server shall sign the Device XML.

Signatures shall follow W3C's recommendation on XML signing, as specified in [3]. The W3C recommendation contains many optional elements for handling the different aspects of the XML signing. In order to reduce the complexity, the following requirements shall be followed for MirrorLink Server and Client devices:

- The **Reference URI** shall be empty.
- MirrorLink Server shall not use XPath or XSLT **XML transformations.**
- The MirrorLink Server shall use **Canonical method** XML version 1.0 (`xml-c14n` or `xml-exc-c14n`) or 1.1 (`xml-c14n11`); Canonical XML version 2.0 or later shall not be used.
- The MirrorLink Server shall use SHA-1 **digest method**; other digest methods shall not be used.
- The MirrorLink Server shall use RSA-SHA1 **signature method**; other signature methods, like HMAC-SHA1 or DSA-SHA1, shall not be used.
- The MirrorLink Client shall not use the ***KeyInfo*** element to identify a public key to verify the signature; it shall use the *applicationPublicKey* element obtained from the DAP *attestationResponse*, for the `TerminalMode:UPnP-Server` component instead.

5 XML Device Description

When processing XML, MirrorLink Clients shall ignore any unknown elements and their sub elements or content and any unknown attributes and their values. MirrorLink Clients shall not expect any particular order of XML elements located at the same level of the XML tree, unless specifically mandated (e.g. via xs:sequence). MirrorLink Client shall understand xml namespace as specified in W3C "Namespaces in XML 1.0".

The MirrorLink Server shall provide a well-formed XML and a correct parent children relationship of xml elements and correct xml namespaces URI for each element. If the MirrorLink Server provides MirrorLink extension elements (X_...) then those elements shall be valid according their XSD provided in Annex A. The MirrorLink Server shall provide XML Device Description that is valid according to XSD presented in Annex A.

The MirrorLink Client should be as permissive as possible when consuming XML.

As a general recommendation for better interoperability, the MirrorLink Servers should declare XML namespaces as default to allow interoperability with Clients without XML namespaces support. MirrorLink Servers shall add any element unspecified below after the specified ones and put it in a namespace that is not specified below. MirrorLink Clients should be able to handle elements in wrong or any xml namespace if the element was not found in the correct one.

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
    <specVersion>
        <major>1</major>
        <minor>0</minor>
```

```
</specVersion>
<URLBase>base URL for all relative URLs</URLBase>
<device>
    <deviceType>
        urn:schemas-upnp-org:device:TmServerDevice:1
    </deviceType>
    <friendlyName>short user-friendly title</friendlyName>
    <manufacturer>manufacturer name</manufacturer>
    <manufacturerURL>URL to manufacturer site</manufacturerURL>
    <modelDescription>long user-friendly title</modelDescription>
    <modelName>model name</modelName>
    <modelNumber>model number</modelNumber>
    <modelURL>URL to model site</modelURL>
    <serialNumber>manufacturer's serial number</serialNumber>
    <UDN>uuid:UUID</UDN>
    <UPC>Universal Product Code</UPC>
    <iconList>
        <icon>
            <mimetype>image/format</mimetype>
            <width>horizontal pixels</width>
            <height>vertical pixels</height>
            <depth>color depth</depth>
            <url>URL to icon</url>
        </icon>
    </iconList>
    <serviceList>
        <service>
            <serviceType>
                urn:schemas-upnp-org:service:TmApplicationServer:1
            </serviceType>
            <serviceId>
                urn:upnp-org:serviceId:TmApplicationServer1
            </serviceId>
            <SCPDURL>URL to service description</SCPDURL>
            <controlURL>URL for control</controlURL>
            <eventSubURL>URL for eventing</eventSubURL>
        </service>
        <service>
            <serviceType>
                urn:schemas-upnp-org:service:TmClientProfile:1
            </serviceType>
            <serviceId>
                urn:upnp-org:serviceId:TmClientProfile1
            </serviceId>
            <SCPDURL>URL to service description</SCPDURL>
            <controlURL>URL for control</controlURL>
            <eventSubURL>URL for eventing</eventSubURL>
        </service>
        <service>
            <serviceType>
                urn:schemas-upnp-org:service:TmNotificationServer:1
            </serviceType>
            <serviceId>
                urn:upnp-org:serviceId:TmNotificationServer1
            </serviceId>
            <SCPDURL>URL to service description</SCPDURL>
            <controlURL>URL for control</controlURL>
            <eventSubURL>URL for eventing</eventSubURL>
        </service>
    </serviceList>
    <presentationURL>URL for presentation</presentationURL>
    <X_connectivity xmlns="urn:schemas-carconnectivity-org:ml-1-0">
        <bluetooth>
            <bdAddr>
                Bluetooth MAC address of server device as a string
                representing a 48 bit number in hexadecimal format
                (without any 0x prefix).
                If this value is populated then it means that the
                device supports Bluetooth communication
            </bdAddr>
            <startConnection>
                Indicates that server device is able to initiate
                Bluetooth connection to client
            </startConnection>
            <clientBdAddr>
                Bluetooth MAC address of client device as a string
                representing a 48 bit number in hexadecimal format
                (without any 0x prefix).
```

```
                </clientBdAddr>
            </bluetooth>
            <wifi>
                <macAddr>EF3456347AB7</macAddr>
                <ssid>MirrorLinkServer_SSID_Example</ssid>
                <protectionList>
                    <protection>
                        <protocol>WPA</protocol>
                        <passkey>ALDFKJESKJFEILKSDFJE</passkey>
                    </protection>
                </protectionList>
            </wifi>
        </X_connectivity>
        <X_mirrorLinkVersion
        xmlns="urn:schemas-carconnectivity-org:ml-1-1">
            <majorVersion>1</majorVersion>
            <minorVersion>3</minorVersion>
        </X_mirrorLinkVersion>
        <X_Signature xmlns="urn:schemas-carconnectivity-org:ml-1-1">
            <Signature Id="deviceSignature"
            xmlns="http://www.w3.org/2000/09/xmldsig#">
                <SignedInfo>
                    <CanonicalizationMethod
                    Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
                    <SignatureMethod Algorithm=
                    "http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
                    <Reference URI="">
                        <Transforms>
                            <Transform Algorithm=
                            "http://www.w3.org/2006/12/xml-c14n11"/>
                            <Transform Algorithm=
                            "http://www.w3.org/2000/09/xmldsig#
                            enveloped-signature"/>
                        </Transforms>
                        <DigestMethod Algorithm=
                        "http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <DigestValue>
                            dGhpcyBpcyBub3QgYSB
                            zaWduYXR1cmUK...
                        </DigestValue>
                    </Reference>
                </SignedInfo>
                <SignatureValue>...</SignatureValue>
            </Signature>
        </X_Signature>
        <X_presentations xmlns="urn:schemas-carconnectivity-org:ml-1-2">
            <presentation>
                vncu,hsml,wfd
            </presentation>
        </X_presentations>
        <X_mlUiMode xmlns="urn:schemas-carconnectivity-org:ml-1-3">
            <mode>immersive</mode>
            <mode>classic</mode>
        </X_mlUiMode>
    </device>
</root>
```

6 Test

No semantic tests are defined for this device.

Annex A (normative): XSD Schema

A.1 XSD Schema for UDA 1.1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="urn:schemas-upnp-org:device-1-0"
 xmlns="urn:schemas-upnp-org:device-1-0"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:ml1-0="urn:schemas-carconnectivity-org:ml-1-0"
 xmlns:ml1-1="urn:schemas-carconnectivity-org:ml-1-1"
 xmlns:ml1-2="urn:schemas-carconnectivity-org:ml-1-2"
 xmlns:ml1-3="urn:schemas-carconnectivity-org:ml-1-3"
 elementFormDefault="qualified">
<xsd:import schemaLocation="ml1-0.xsd" namespace="urn:schemas-carconnectivity-org:ml-1-0"/>
<xsd:import schemaLocation="ml1-1.xsd" namespace="urn:schemas-carconnectivity-org:ml-1-1"/>
<xsd:import schemaLocation="ml1-2.xsd" namespace="urn:schemas-carconnectivity-org:ml-1-2"/>
<xsd:import schemaLocation="ml1-3.xsd" namespace="urn:schemas-carconnectivity-org:ml-1-3"/>
<xsd:element name="root" type="rootType"/>
 <xsd:complexType name="deviceType">
  <xsd:sequence>
   <xsd:element name="deviceType">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:anyURI">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
    </xsd:complexType>
   </xsd:element>
   <xsd:element name="friendlyName">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:string">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
    </xsd:complexType>
   </xsd:element>
   <xsd:element name="manufacturer" minOccurs="1">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:string">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
    </xsd:complexType>
   </xsd:element>
   <xsd:element name="manufacturerURL" minOccurs="0">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:anyURI">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
    </xsd:complexType>
   </xsd:element>
   <xsd:element name="modelDescription" minOccurs="0">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:string">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
    </xsd:complexType>
   </xsd:element>
   <xsd:element name="modelName">
    <xsd:complexType>
     <xsd:simpleContent>
      <xsd:extension base="xsd:string">
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:extension>
     </xsd:simpleContent>
```

```
 </xsd:complexType>
</xsd:element>
<xsd:element name="modelNumber" minOccurs="1">
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:string">
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>
<xsd:element name="modelURL" minOccurs="0">
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:anyURI">
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>
<xsd:element name="serialNumber" minOccurs="0">
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:string">
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>
<xsd:element name="UDN">
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:anyURI">
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>
<xsd:element name="UPC" minOccurs="0">
 <xsd:complexType>
  <xsd:simpleContent>
   <xsd:extension base="xsd:string">
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
</xsd:element>
<xsd:element name="iconList" minOccurs="0">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="icon" maxOccurs="unbounded">
    <xsd:complexType>
     <xsd:sequence>
      <xsd:element name="mimetype">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:string">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="width">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:int">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="height">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:int">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
```

```
      </xsd:complexType>
     </xsd:element>
     <xsd:element name="depth">
      <xsd:complexType>
       <xsd:simpleContent>
        <xsd:extension base="xsd:int">
         <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
       </xsd:simpleContent>
      </xsd:complexType>
     </xsd:element>
     <xsd:element name="url">
      <xsd:complexType>
       <xsd:simpleContent>
        <xsd:extension base="xsd:anyURI">
         <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
       </xsd:simpleContent>
      </xsd:complexType>
     </xsd:element>
     <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:complexType>
  </xsd:element>
  <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
 </xsd:sequence>
 <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
</xsd:element>
<xsd:element name="serviceList">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="service" maxOccurs="unbounded">
    <xsd:complexType>
     <xsd:sequence>
      <xsd:element name="serviceType">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:anyURI">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="serviceId">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:anyURI">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="SCPDURL">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:anyURI">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="controlURL">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:anyURI">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
        </xsd:simpleContent>
       </xsd:complexType>
      </xsd:element>
      <xsd:element name="eventSubURL">
       <xsd:complexType>
        <xsd:simpleContent>
         <xsd:extension base="xsd:anyURI">
          <xsd:anyAttribute namespace="##other" processContents="lax"/>
         </xsd:extension>
```

```
          </xsd:simpleContent>
         </xsd:complexType>
        </xsd:element>
        <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
       </xsd:sequence>
       <xsd:anyAttribute namespace="##other" processContents="lax"/>
      </xsd:complexType>
     </xsd:element>
     <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="deviceList" type="deviceListType" minOccurs="0"/>
  <xsd:element name="presentationURL" minOccurs="0">
   <xsd:complexType>
    <xsd:simpleContent>
     <xsd:extension base="xsd:anyURI">
      <xsd:anyAttribute namespace="##other" processContents="lax"/>
     </xsd:extension>
    </xsd:simpleContent>
   </xsd:complexType>
  </xsd:element>
  <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  <xsd:element minOccurs="0" ref="ml1-0:X_connectivity"/>
  <xsd:element minOccurs="0" ref="ml1-0:X_deviceKeys"/>
  <xsd:element minOccurs="0" ref="ml1-1:X_mirrorLinkVersion"/>
  <xsd:element minOccurs="1" ref="ml1-1:X_Signature"/>
  <xsd:element minOccurs="1" ref="ml1-2:X_presentations"/>
  <xsd:element minOccurs="0" ref="ml1-2:X_localization"/>
  <xsd:element minOccurs="1" ref="ml1-3:X_mlUiMode"/>
  <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
 </xsd:sequence>
 <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="deviceListType">
 <xsd:sequence>
  <xsd:element name="device" type="deviceType" maxOccurs="unbounded"/>
  <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
 </xsd:sequence>
 <xsd:anyAttribute namespace="##other" processContents="lax"/>
</xsd:complexType>
<xsd:complexType name="rootType">
 <xsd:sequence>
  <xsd:element name="specVersion">
   <xsd:complexType>
    <xsd:sequence>
     <xsd:element name="major">
      <xsd:complexType>
       <xsd:simpleContent>
        <xsd:extension base="xsd:int">
         <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
       </xsd:simpleContent>
      </xsd:complexType>
     </xsd:element>
     <xsd:element name="minor">
      <xsd:complexType>
       <xsd:simpleContent>
        <xsd:extension base="xsd:int">
         <xsd:anyAttribute namespace="##other" processContents="lax"/>
        </xsd:extension>
       </xsd:simpleContent>
      </xsd:complexType>
     </xsd:element>
     <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
   </xsd:complexType>
  </xsd:element>
  <xsd:element name="URLBase" minOccurs="0">
   <xsd:complexType>
    <xsd:simpleContent>
     <xsd:extension base="xsd:anyURI">
      <xsd:anyAttribute namespace="##other" processContents="lax"/>
     </xsd:extension>
    </xsd:simpleContent>
   </xsd:complexType>
```

```
   </xsd:element>
   <xsd:element name="device" type="deviceType"/>
   <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xsd:sequence>
  <xsd:attribute name="configId" type="xsd:int"/>
  <xsd:anyAttribute namespace="##other" processContents="lax"/>
 </xsd:complexType>
</xsd:schema>
```

A.2 ml1-0.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-carconnectivity-org:ml-1-0"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-carconnectivity-org:ml-1-0"
elementFormDefault="qualified">
<xs:element name="X_connectivity">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="bluetooth" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="bdAddr" type="xs:string" minOccurs="1"/>
      <xs:element name="startConnection" type="xs:boolean" minOccurs="0" default="true"/>
      <xs:element name="clientBdAddr" type="xs:string" minOccurs="0"/>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:element name="wifi" minOccurs="0">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="macAddr" type="xs:string" minOccurs="1"/>
      <xs:element name="ssid" type="xs:string" minOccurs="0"/>
      <xs:element name="roles" type="xs:string" minOccurs="0"
       default="AP,Client,P2P"/>
      <xs:element name="protectionList" minOccurs="0">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="protection" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
           <xs:sequence>
            <xs:element name="protocol" minOccurs="1">
             <xs:simpleType>
              <xs:restriction base="xs:string">
               <xs:enumeration value="WEP"/>
               <xs:enumeration value="WPA"/>
               <xs:enumeration value="WPA2"/>
               <xs:enumeration value="WPS"/>
              </xs:restriction>
             </xs:simpleType>
            </xs:element>
            <xs:element name="passkey" type="xs:string" minOccurs="1"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
           </xs:sequence>
           <xs:anyAttribute namespace="##any" processContents="lax"/>
          </xs:complexType>
         </xs:element>
         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
       </xs:complexType>
      </xs:element>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="X_deviceKeys">
 <xs:complexType>
```

```
  <xs:sequence>
   <xs:element name="key" minOccurs="0" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="name" type="xs:string" minOccurs="1"/>
      <xs:element name="mandatory" type="xs:boolean" minOccurs="0" default="false"/>
      <xs:element name="symbolValue" type="xs:string" minOccurs="1"/>
      <xs:element name="icon" maxOccurs="unbounded">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="mimetype" type="xs:string" minOccurs="1"/>
         <xs:element name="width" type="xs:positiveInteger" minOccurs="1"/>
         <xs:element name="height" type="xs:positiveInteger" minOccurs="1"/>
         <xs:element name="depth" type="xs:positiveInteger" minOccurs="1"/>
         <xs:element name="url" type="xs:string" minOccurs="1"/>
         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
       </xs:complexType>
      </xs:element>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
     </xs:sequence>
     <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
   </xs:element>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
</xs:schema>
```

A.3 ml1-1.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-carconnectivity-org:ml-1-1"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-carconnectivity-org:ml-1-1"
elementFormDefault="qualified" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<xs:import schemaLocation="xmldsig-core-schema.xsd"
 namespace="http://www.w3.org/2000/09/xmldsig#"/>
<xs:element name="X_mirrorLinkVersion">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="majorVersion" type="xs:nonNegativeInteger"/>
   <xs:element name="minorVersion" type="xs:nonNegativeInteger"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="X_Signature" minOccurs="1">
 <xs:complexType>
  <xs:sequence>
   <xs:element ref="ds:Signature" minOccurs="1"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
</xs:schema>
```

A.4 ml1-2.xsd (MirrorLink 1.2)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-carconnectivity-org:ml-1-2"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-carconnectivity-org:ml-1-2"
elementFormDefault="qualified">
<xs:element name="X_presentations">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="presentation" type="xs:string" default="vncu" minOccurs="1"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
```

```
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
<xs:element name="X_localization" minOccurs="0">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="characterSet" type="xs:string" minOccurs="1"/>
   <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
</xs:schema>
```

A.5 ml1-3.xsd (MirrorLink 1.3)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:schemas-carconnectivity-org:ml-1-3"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:schemas-carconnectivity-org:ml-1-3"
elementFormDefault="qualified">
<xs:element name="X_mlUiMode">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="mode" minOccurs="1" minOccurs="unbounded">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="immersive"/>
      <xs:enumeration value="classic"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
  </xs:sequence>
  <xs:anyAttribute namespace="##any" processContents="lax"/>
 </xs:complexType>
</xs:element>
</xs:schema>
```

Annex B (informative): Authors and Contributors

The following people have contributed to the present document:

Rapporteur:	Dr. Jörg Brakensiek, E-Qualus (for Car Connectivity Consortium LLC)
Other contributors:	Raja Bose, Nokia Corporation

History

Document history		
V1.3.0	October 2017	Publication
V1.3.1	October 2019	Publication