ETSI TS 103 544-24 V1.3.0 (2017-10)

Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 24: Media Meta Data Service

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
DTS/ITS-88-24

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 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
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 24 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 colour display, audio input/output and user input mechanisms.

Current MirrorLink solutions are concentrated on utilization of MirrorLink Client's main display to mirror applications or provide variety services on the MirrorLink Server. However, there are so many MirrorLink Clients which have several other displays, such as cluster display panel, Heads-up Display (HUD) and so on. Instead of applications mirroring, using these displays, the driver and the passenger can be provided with a variety meta information such as turn by turn information, photo or graphic information, meta data information of audio and video clip, text information, etc. Those Meta Information Data Services are based on the SBP (Service Binary Protocol) framework.

The present document specifies media meta data service based on SBP (Service Binary Protocol) framework. By receiving this data, the MirrorLink Client (e.g. a car) can provide media information to driver and passenger e.g. through the car's cluster display panel, or heads-up display.

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] ETSI TS 103 544-27 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 27: Basic Meta Data Service".

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 (V1.3.0): "Publicly Available Specification (PAS); Intelligent Transport Systems (ITS); MirrorLink®; Part 1: Connectivity".

3 Abbreviations

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

H/U	Head Unit
HUD	Head Up Display

4 Data Service Definition

4.1 Media Meta Data Service Version 1.0

```
/** The present document defines data objects for the Media Meta
 *  data service to be carried over the SBP. By receiving
 *  this data, the Media Data Sink (e.g. a car) can provide variety
 *  media information to the driver and passenger through instrument
 *  cluster display panel, HUD etc.
 *  The Media Meta Source may be implemented from a MirrorLink Client
 *  or a MirrorLink Server (typically driven from a media player
 *  application).
 *  The service is based on the Basic Meta Data Service.
 *  @version 1.0
 */
SERVICE com.mirrorlink.meta.media
  : com.mirrorlink.meta.basic version 1.0 {
/** Enumeration describing the status of the media player.
 */
ENUM<BYTE> PlayStatus {
  /** Media playback stopped. No audio playback. The Media player shall
   *  NOT stream audio to the Media Sink.
   */
  STOPPED = 0x00,
  /** Prepare Media playback. Media player is preparing the media
   *  stream. A media player may skip this state and immediately go
   *  to the PLAYING state. The Media player shall not stream audio to
   *  the Media Sink.
   */
  PREPARING = 0x01,
  /** Media playback ongoing. In PLAYING state, the Media player shall
   *  stream audio to the Media Sink.
   */
  PLAYING = 0x02,
  /** Media playback paused. Media player typically switches to PAUSED
   *  state based on user action. In PAUSED state, the Media player
   *  shall not stream audio to the Media Sink.
   */
  PAUSED = 0x03
  };
/** Enumeration describing the commands for the media player.
 */
ENUM<BYTE> Command {
  /** Start media playback.
   */
  START = 0x00,
  /** Stop media playback.
   */
  STOP = 0x01,
  /** Pause media playback.
   */
  PAUSE = 0x02,
  /** Start playback of the next media item from the browsable
   *  media list.
   */
  NEXT = 0x03,
  /** Start playback of the previous media item from the browsable
   *  media list. This command may go back to the start position of
   *  the current media item first. A second call will then go to the
   *  previous track. The behavior is media player specific.
   */
  PREVIOUS = 0x04
  /** Fast forward.
   */
```

```
  FORWARD = 0x05,
  /** Fast backward.
   */
  REWIND = 0x06
  };
/** The MediaItem structure holds all meta data related to a specific
 *  media item (e.g. a song, a play list item).
 */
STRUCTURE MediaItem {
  /** Duration of the media item's playtime, if it would be played in
   *  full. The value is representing milliseconds [ms].
   *  @optional, @unit milliseconds, @uid 0xb62c9639
   */
  INT duration;

  /** Title of the media item. In case the title is not known, it shall
   *  be set to an empty String.
   *  Shall not exceed 60 characters.
   *  @mandatory, @uid 0x28c9dd73
   */
  STRING title;
  /** Artist of the media item. In case the artist is not known, it
   *  Shall be set to an empty String.
   *  Shall not exceed 60 characters.
   *  @mandatory, @uid 0xa3b8740c
   */
  STRING artist;
  /** Album title the media item belongs to.
   *  Shall not exceed 60 characters.
   *  @optional, @uid 0xa0ff2dca
   */
  STRING album;
  /** Year the media item has been published/recorded.
   *  Shall not exceed 4 characters.
   *  @optional, @uid 0xe73ed082
   */
  STRING year;
  /** Genre of the media item.
   *  Shall not exceed 30 characters.
   *  @optional, @uid 0xf8bf6bde
   */
  STRING genre;
  /** Mimetype of the media item artwork.
   *  Shall be included, if artwork is provided. Artwork should be
   *  either "image/png" or "image/jpeg".
   *  @conditional, @uid 0x107e35f3
   */
  STRING mimetype;

  /** Artwork of the media item.
   *  @optional, @uid 0x47d5bb4f
   */
  BYTES artwork;
  };
/** The MediaPlayerStatus object informs the data sink of the media
 *  player and its status.
 *  @mandatory, @readable, @uid 0x5336a7b2
 */
OBJECT MediaPlayerStatus {
  /** Identifier of the media player application, as used within UPnP
   *  application advertisements and the respective framebuffer
   *  context information.
   *  Shall be set to 0, if no media player is running and registered
   *  to the media meta data service.
   *  @mandatory, @uid 0xcc99c513
   */
  INT playingAppId;
  /** Status of the media player.
   *  @mandatory, @uid 0x3a2627d2
   */
  ENUM<PlayStatus> mediaPlayerStatus;
  };
/** The CurrentMedia object inform the data sink about the currently
 *  playing media item.
 *  @mandatory, @readable, @version 1.0, @uid 0x954b7043
 */
OBJECT CurrentMediaItem {
  /** Currently playing media item.
```

```
   *  @mandatory, @uid 0x49d9e823
   */
  STRUCTURE<MediaItem> currentMediaItem;
  };
/** The NextMediaItem object inform the data sink about the playing
 *  media item, which will be played next.
 *  @optional, @readable, @version 1.0, @uid 0x7fe6e95f
 */
OBJECT NextMediaItem {
  /** Currently playing media item.
   *  @mandatory, @uid 0x10ea897f
  STRUCTURE<MediaItem> nextMediaItem;
  };
/** The PlayMediaItem object allows the data sink to set the media
 *  item to be played from the Media Player.
 *  @optional, @writable, @version 1.0, @uid 0x05f6417f
 */
OBJECT ControlMediaItem {
  /** Identifier of the command, which should be executed on the given
   *  media item.
   *  The following commands shall be supported:
   *   - START, PAUSE, STOP,
   *   - NEXT, PREVIOUS.
   *  The following commands should be supported:
   *   - REWIND, FORWARD. If one command is supported, the other shall
   *     be supported as well.
   *  @mandatory, @uid 0xfe02da06
   */
  ENUM<Command> command;
  };
/** The PlaybackTime object informs the data sink of the currentMedia
 *  playback timing. Synchronization with currentMedia item cannot be
 *  guaranteed 100%. Update to PlaybackPosition object may happen
 *  prior or after the update of the currentMedia item.
 *  @optional, @uid 0x21cac1c9
 */
OBJECT PlaybackPosition {
  /** Already played time for the currently playing media item.
   *  Time duration is provided in milliseconds.
   *  @mandatory, @unit milliseconds, @uid 0xec996318
   */
  INT played;
  };
};
```

5 SBP Binding

5.1 Service Registry

The Media Meta Data Services uses the following objects and their access capabilities, as defined in [1].

Table 1

name / uid	accessType	subscriptionType	minIntervalTime	MaxIntervalTime
MediaPlayerStatus	READABLE	ON_CHANGE	N/A	N/A
CurrentMediaItem	READABLE	ON_CHANGE	N/A	N/A
NextMediaItem	READABLE	ON_CHANGE	N/A	N/A
ControlMediaItem	WRITABLE	NONE	N/A	N/A
PlaybackPosition	READABLE	REGULAR or AUTO	≤ 1 s	≥ 10 s

5.2 Service Specific Error Codes

5.2.1 General

The Media Meta data service defines the following service specific error codes, used within an *SBP::Response*.

The Media Meta data service shall not terminate in response to any service-specific error. The Media Meta data service source shall send the respective error response, but shall otherwise ignore the command.

5.2.2 Access to ControlMediaItem Object

The following error codes are defined, accessing the *ControlMediaItem* object:

`0x40000 0000`: *command* not known.
This error code is sent, when the sink selects a media item not defined in the enumeration.

`0x40000 0001`: *command* not supported.
This error code is sent, when the sink is sending a command not supported from the Media Meta Data Service Source.

`0x40000 0002`: *command* failed.
This error code is sent, when the execution of the *command* failed.

6 Theory of Operation

6.1 Subscribing to Current and Next Media Items

Figure 1 shows how a media data sink retrieves meta data about the currently played media from the media data source and displays it e.g. on the Instrument Cluster Display (ICD).

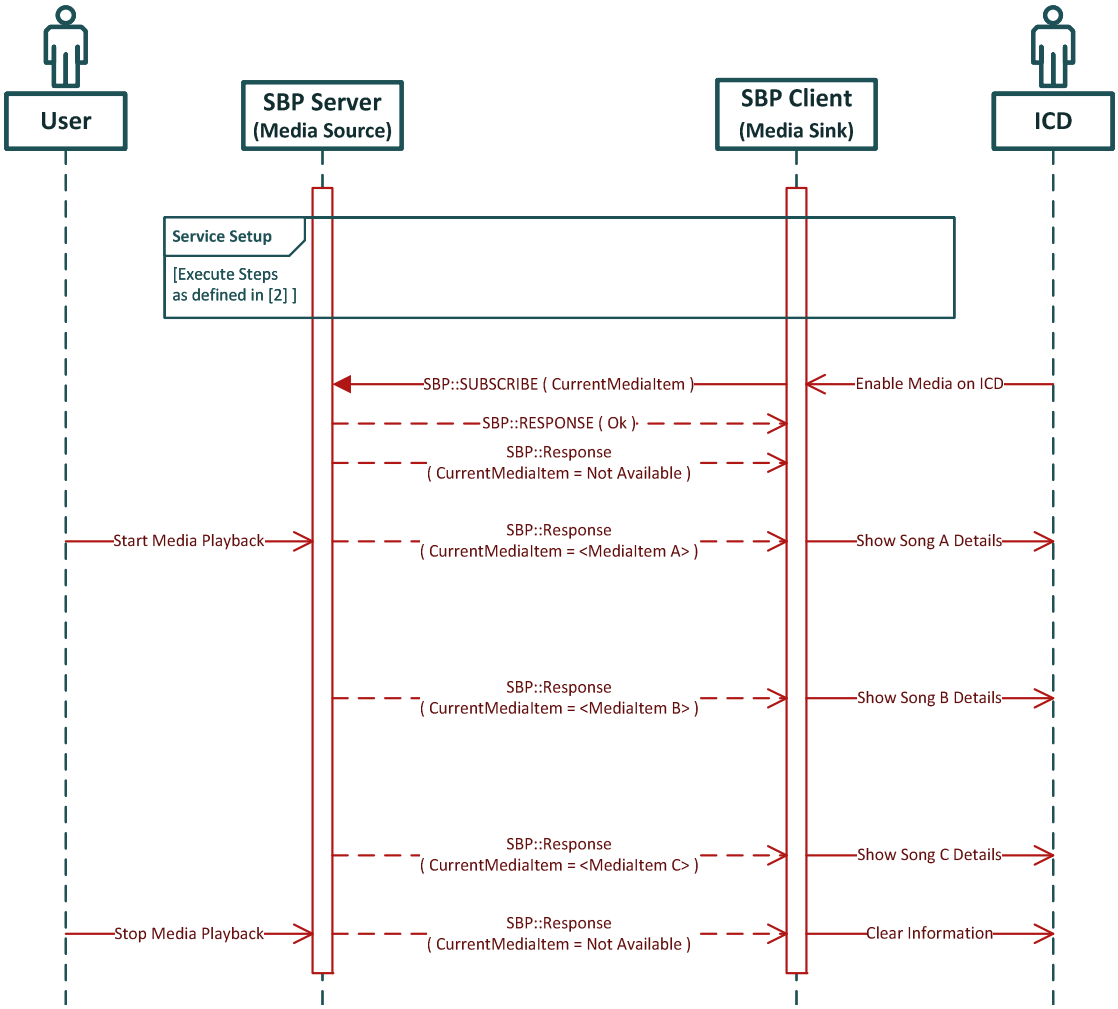

Figure 1: Message Sequence Diagram - Retrieve Media Meta Data of currently played Media

It consists of the following steps, after the data service has been setup as defined in [1]:

1) Media Sink sends an SBP *Subscribe* message for the *CurrentMediaItem* object; subscription is ON_CHANGE. The Media Source responds first with an SBP *Response* message confirming the *Subscribe* message, followed by a second SBP *Response* message containing an `"Not Available"` error code, as no media is currently played.

NOTE: In case media playback started prior subscription, the Media Sink will return a valid media item object, representing the current media item.

2) The user is starting the media playback.

3) Media Sink sends an SBP *Response* message for the *CurrentMediaItem*, containing the media item object, representing the currently played media item.

4) Media Sink sends another SBP *Response* message for the *CurrentMediaItem*, containing the media item object, whenever the media player goes to the next media item.

5) The user is stopping the media playback.

6) Media Sink sends an SBP *Response* message for the *CurrentMediaItem*, containing an `"Not Available"` error code, as no media is currently played.

The Media Sink can subscribe to the *NextMediaItem* as well, which will contain the information of the next, upcoming media item in the playlist. For some media types, like streaming media, the *NextMediaItem* object will not be available, leading to the error code `"Not Available"`. When the media player switches to the next track, both objects, *CurrentMediaItem* and *NextMediaItem* will get updated. There is no specific order for the update.

6.2 Subscribing to Playback Position

Figure 2 shows how a media data sink retrieves meta data about the current playback position, for the current media playback, from the media data source and displays it e.g. on the Instrument Cluster Display (ICD).

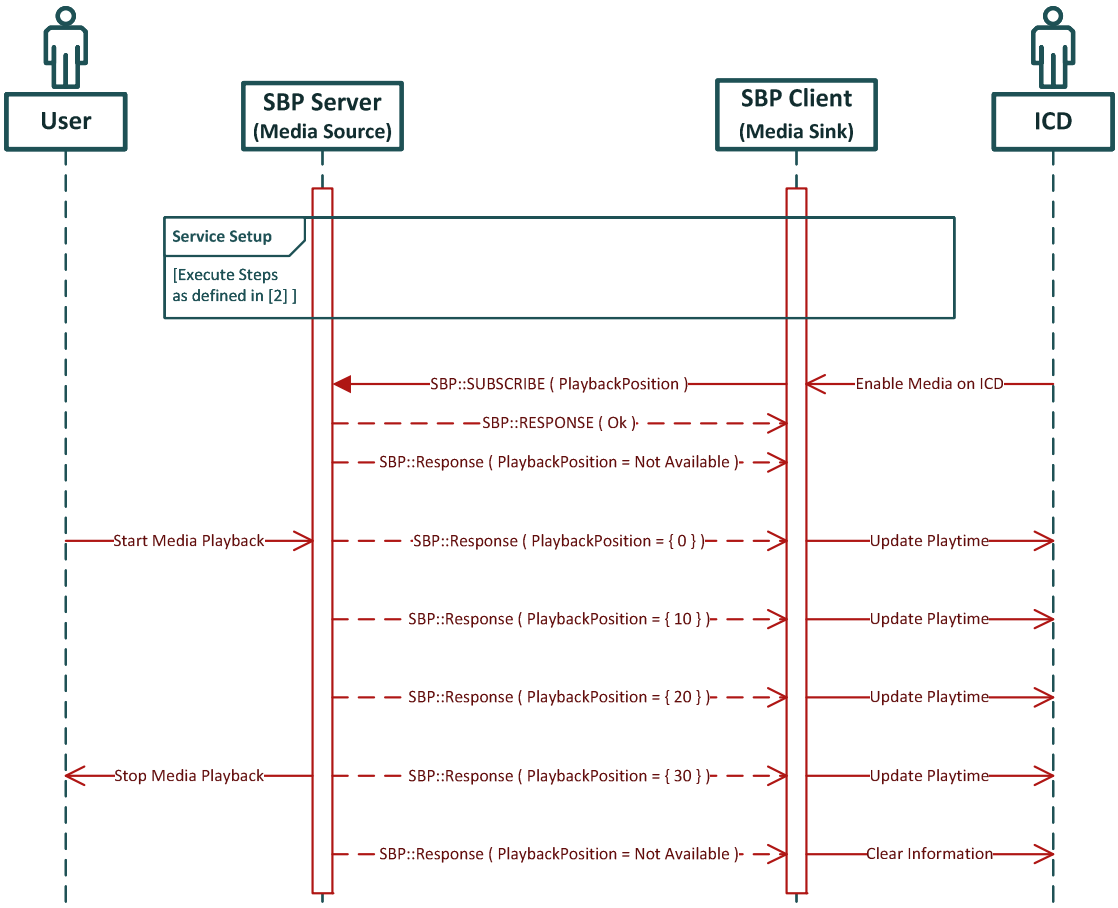

Figure 2: Message Sequence Diagram - Retrieve Media Meta Data of currently played Media

It consists of the following steps, after the data service has been setup as defined in [1]:

1) Media Sink sends an SBP *Subscribe* message for the *PlaybackPosition* object; subscription is REGULAR. The Media Source responds first with an SBP *Response* message confirming the *Subscribe* message, followed by a second SBP *Response* message containing an `"Not Available"` error code, as no media is currently played.

NOTE: In case media playback started prior subscription, the Media Sink will return a valid media item object, representing the current media item.

2) The user is starting the media playback.

3) Media Sink sends an SBP *Response* message for the *PlaybackPosition*, containing two time values (in ms), the *played* time and the *remaining* time. Note that despite the value being provided in ms, the accuracy is not necessarily at ms level, as the SBP is subject to network jitter and the transfer is not synchronized with the ongoing audio stream.

4) Media Sink sends another SBP *Response* message for the *PlaybackPosition*, containing the *played* time and the *remaining* time, at the requested regular updates.

5) The media playback stops at the end of the media item.

6) Media Sink sends an SBP *Response* message for the *PlaybackPosition*, containing an `"Not Available"` error code, as no media is currently played.

In case the Media Data Sink has subscribed to the *CurrentMediaItem* object and the *PlaybackPosition* object, one or the other objects is updated first, when the Media Data Source starts the playback of a new media item.

6.3 Remote Control the Media Source

Figure 3 shows how a media data sink can remote control the media data source. To support the remote-control mechanisms, the Media Data Source is required to support the *ControlMediaItem* object. An example message sequence diagram is shown in Figure 3.

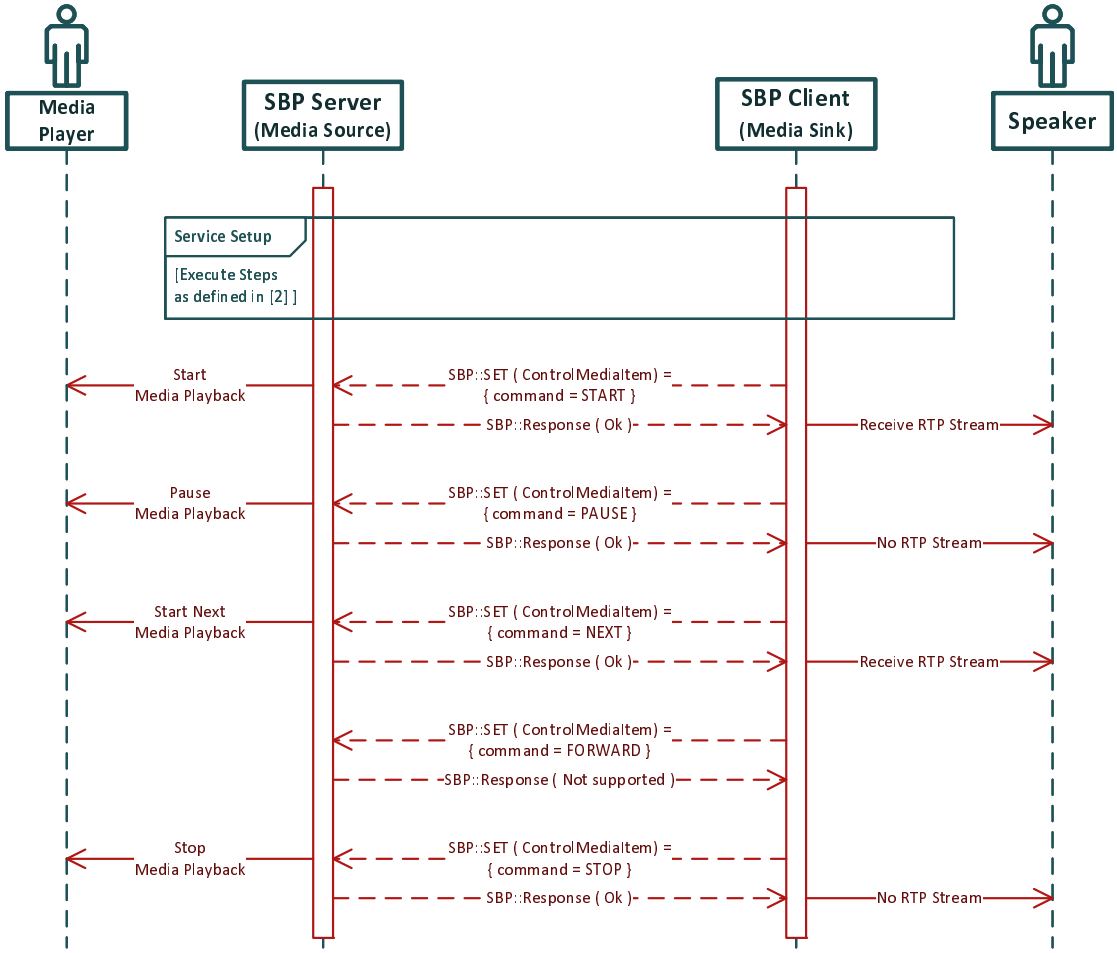

Figure 3: Message Sequence Diagram - Remote Control

It consists of the following steps, after the data service has been setup as defined in [1]:

1) To **start** the playback, the Media Meta Data Sink sends an SBP *Set* message for the *ControlMediaItem* object, containing the PLAY command.

2) The Media Meta Data Sink starts the media playback and sends an SBP *Response* Ok message. The RTP audio stream is received from the RTP Client of the MirrorLink device.

3) To **pause** the playback, the Media Meta Data Sink sends an SBP *Set* message for the *ControlMediaItem* object, containing the PAUSE command.

4) The Media Meta Data Sink pauses the media playback and sends an SBP *Response* Ok message. The RTP audio stream is paused.

5) To **start** the playback of the **next** media item in the playlist, the Media Meta Data Sink sends an SBP *Set* message for the *ControlMediaItem* object, containing the NEXT command.

6) The Media Meta Data Sink starts the media playback of the next media item and sends an SBP *Response* Ok message. The RTP audio stream is received from the RTP Client of the MirrorLink device.

7) To **fast forward** the playback of current media item, the Media Meta Data Sink sends an SBP *Set* message for the *ControlMediaItem* object, containing the FORWARD command.

8) The Media Meta Data Sink does not support the command (in this example) and ignores it. It sends an SBP *Response* Error message with error code `"command not supported"`. The RTP audio stream is continues without interruption.

9) To **stop** the playback, the Media Meta Data Sink sends an SBP *Set* message for the *ControlMediaItem* object, containing the STOP command.

10) The Media Meta Data Sink stops the media playback and sends an SBP *Response* Ok message. The RTP audio stream is stopped.

In case the MirrorLink Sink is utilizing the Remote-Control features of the data service it shall not send any Multimedia key events via VNC or WFD UIBC.

Annex A (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:	ARam Cho, Samsung Electronics
	Kiran Vedula, Samsung Electronics
	Inyoung Shin, Samsung Electronics
	Mayur, Samsung Electronics
	Matthias Benesch, Daimler AG
	Gautier Falconnet, PSA
	Laurent Cremmer, RealVNC Ltd.

History

Document history		
V1.3.0	October 2017	Publication