ETSI ES 202 504-11 V1.1.1 (2008-05)

ETSI Standard

Open Service Access (OSA);
Parlay X Web Services;
Part 11: Audio Call
(Parlay X 3)

Reference

DES/TISPAN-01034-11-OSA

Keywords

API, OSA, service

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

Copyright Notification

Contents

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://webapp.etsi.org/IPR/home.asp).

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 ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 11 of a multi-part deliverable covering Open Service Access (OSA); Parlay X Web Services, as identified below:

Part 1: "Common";

Part 2: "Third Party Call";

Part 3: "Call Notification";

Part 4: "Short Messaging";

Part 5: "Multimedia Messaging";

Part 6: "Payment";

Part 7: "Account Management";

Part 8: "Terminal Status";

Part 9: "Terminal Location";

Part 10: "Call Handling";

Part 11: "Audio Call";

Part 12: "Multimedia Conference";

Part 13: "Address List Management";

Part 14: "Presence";

Part 15: "Message Broadcast";

Part 16: "Geocoding";

Part 17: "Application-driven Quality of Service (QoS)";

Part 18: "Device Capabilities and Configuration";

Part 19: "Multimedia Streaming Control";

Part 20: "Multimedia Multicast Session Management".

The present document has been defined jointly between ETSI, The Parlay Group (http://www.parlay.org) and the 3GPP.

The present document forms part of the Parlay X 3.0 set of specifications.

The present document is equivalent to 3GPP TS 29.199-11 V7.2.0 (Release 7).

1 Scope

The present document is part 11 of the Stage 3 Parlay X 3 Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs.

The present document specifies the Call Handling Web Service. The following are defined here:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service Policies.
- WSDL Description of the interfaces.

The web service had been extended to support media.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
    - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
    - for informative references.

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

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

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

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[2] ETSI ES 202 504-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 3)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 504-1 [2] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 202 504-1 [2] apply.

4 Detailed service description

The web service provides a flexible way to provide multimedia message delivery and the dynamic management of the media involved for the call participants. The interface is very simple, not requiring the developer to manage the creation of the call.

The Audio Call web service allows media to be added/dropped for any ongoing call. This web service also allows interaction with other call control web services (e.g. multimedia conference, third party call), enabling delivery of multimedia to call participants in an ongoing call.

The underlying model of the service is based on the following entities:

- **Call Session:** a call (uniquely identified) to which participants can be added/removed.
- **Call Participant:** each of the call parties (uniquely identified) involved in the call session.
- **Media:** the call can utilize multiple media types to support the participants' communication. In particular both audio and video streams are available, including the specific stream direction (i.e. incoming, outgoing, bidirectional).

NOTE: Call participants in a Call Session are anticipated to be uniquely identifiable using their URI address.

There are several mechanisms which may be utilized for the message content:

- Text, to be rendered using a Text-To-Speech (TTS) engine.
- Audio content (such as .WAV content), to be rendered by an audio player.
- VoiceXML, to be rendered using a VoiceXML browser.
- Video, to provide video streaming to the user.
- Capture media input from the end user.

The service may provide one or more of these mechanisms, as determined by service policy.

The service allows application control of the call participants' multimedia in a call:

- Allow multiple media types for each participant. In particular both audio and video as well as chat and data.
- Add and delete media types.
- Control the specific media stream direction (i.e. incoming, outgoing, bidirectional) for each media type.
- Get the current media status of a single call participant or for all the call participants in a call.
- Control the media interactions for a call participant.

A service policy determines if multimedia application control is supported.

5 Namespaces

The PlayMedia interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/audio_call/play_media/v3_2

The CaptureMedia interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/audio_call/capture_media/v3_1

The Multimedia interface uses the namespace:

http://www.csapi.org/wsdl/parlayx/audio_call/multimedia/v3_1

The data types are defined in the namespace:

http://www.csapi.org/schema/parlayx/audio_call/v3_2

The "xsd" namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. The use of the name "xsd" is not semantically significant.

6 Sequence diagrams

6.1 Play audio and check status

Pattern: Request / response.

This example shows an audio message being played, and the different responses to status requests that occur at different phases. Note that the last response, a service exception, reflects the transient nature of results, and that these results will expire.

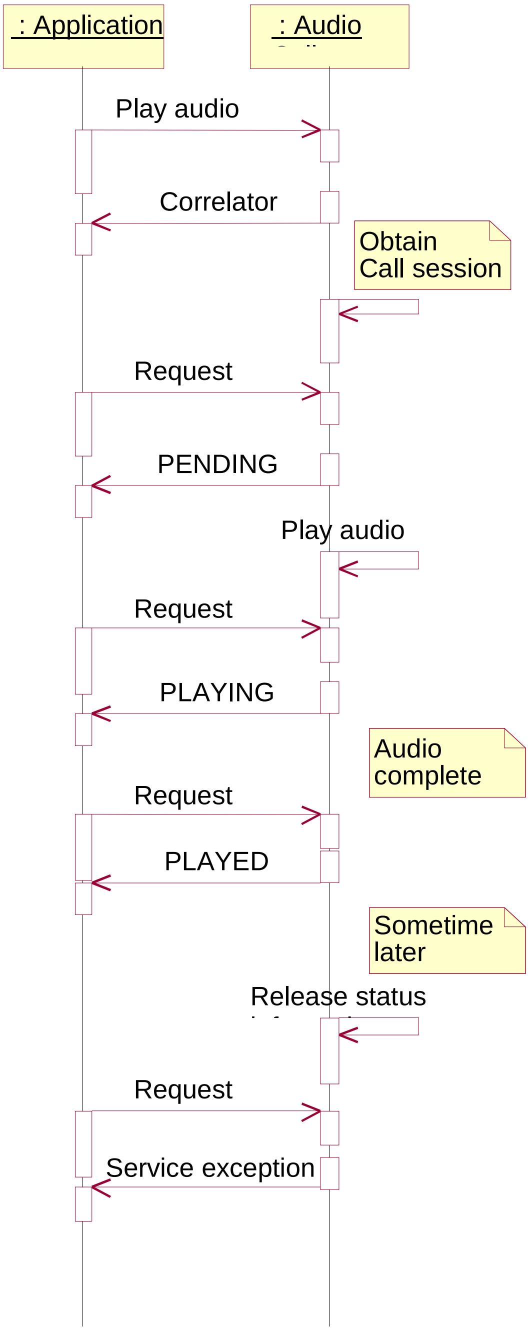
: Application
: Audio
Play audio
Correlator
Obtain Call session
Request
PENDING
Play audio
Request
PLAYING
Audio complete
Request
PLAYED
Sometime later
Release status
Request
Service exception

Figure 1

6.2 Play audio and cancel

Pattern: Request / response.

The playing of a message may be ended by the requester, as shown.

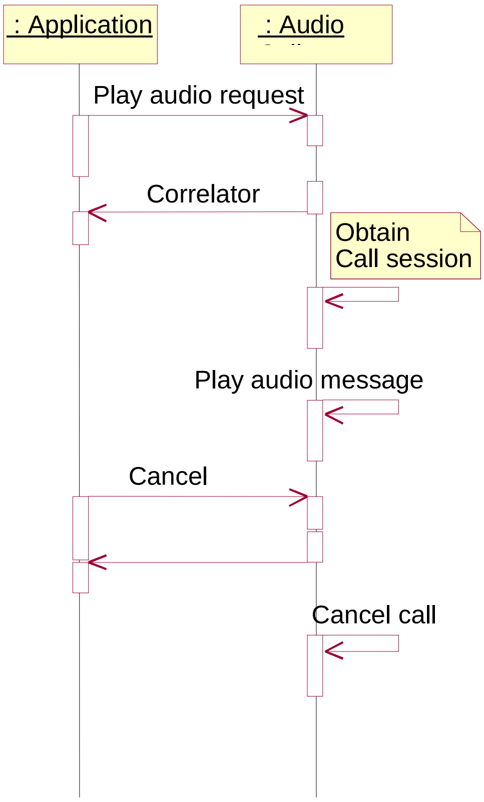

Figure 2

6.3 Adding and removing media

Pattern: Request / response.

This example shows how to add media, read media information, and remove media for a participant on an existing call.

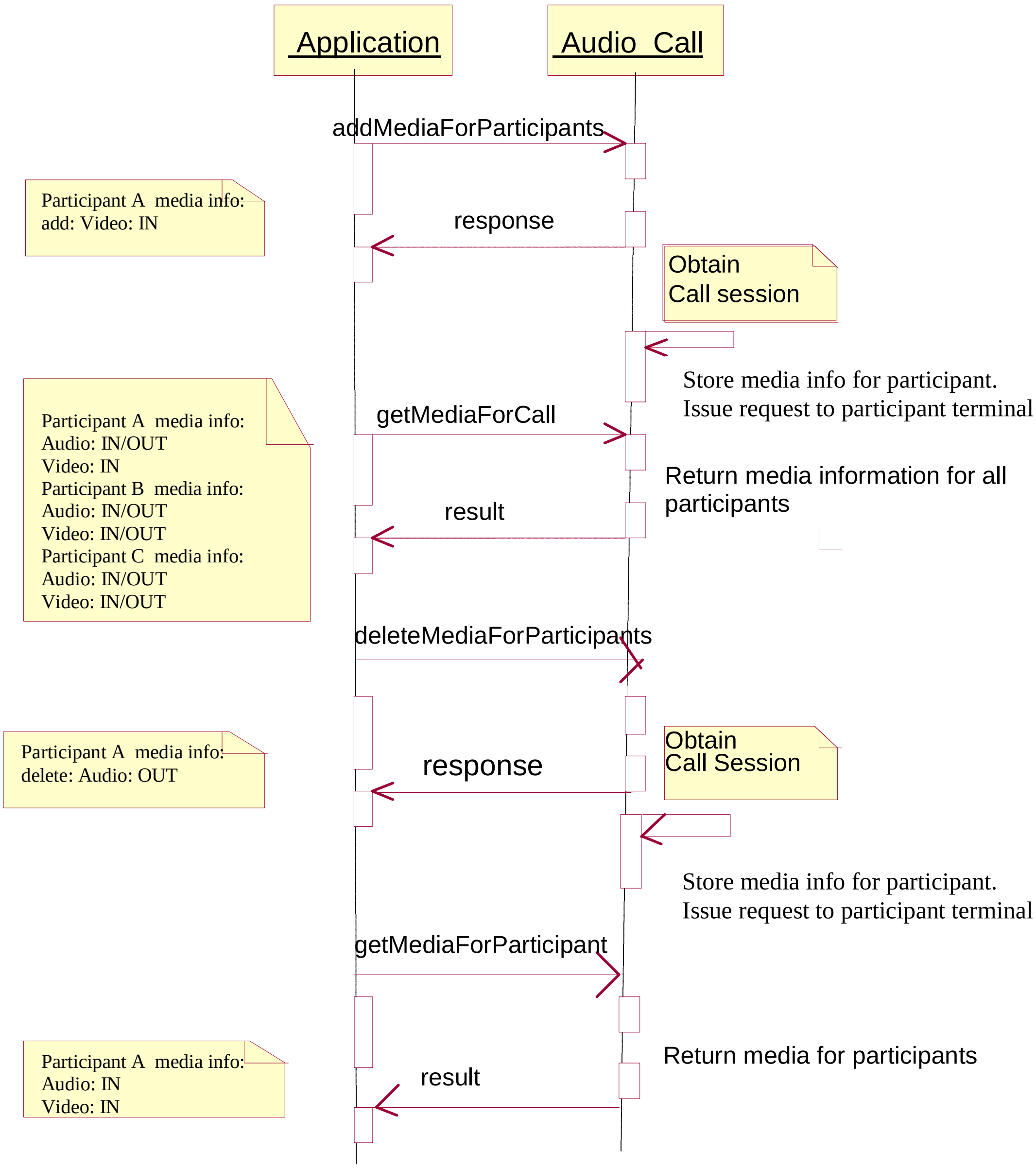
Application
Audio Call
addMediaForParticipants
Participant A media info:
add: Video: IN
response
Obtain
Call session
Store media info for participant.
Issue request to participant terminal
Participant A media info:
Audio: IN/OUT
Video: IN
Participant B media info:
Audio: IN/OUT
Video: IN/OUT
Participant C media info:
Audio: IN/OUT
Video: IN/OUT
getMediaForCall
Return media information for all participants
result
deleteMediaForParticipants
Participant A media info:
delete: Audio: OUT
response
Obtain
Call Session
Store media info for participant.
Issue request to participant terminal
getMediaForParticipant
Participant A media info:
Audio: IN
Video: IN
Return media for participants
result

Figure 3

6.4 Play media file and collect digits from end user

This example shows an application playing a media file and retrieving digits from the end user.

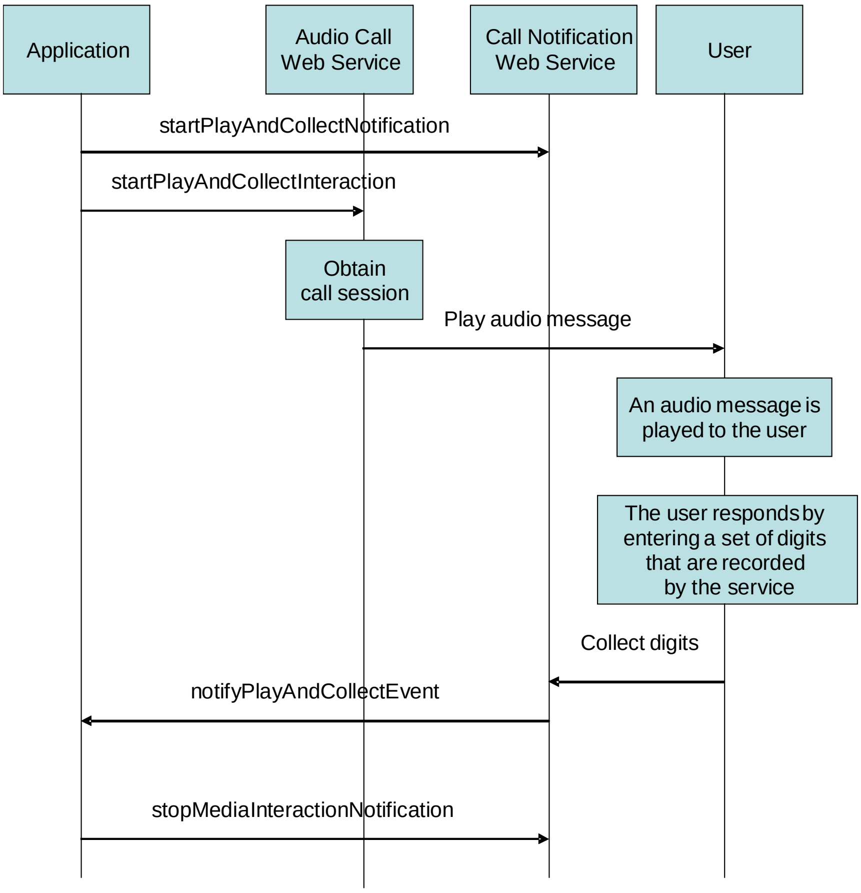
Application
Audio Call Web Service
Call Notification Web Service
User
startPlayAndCollectNotification
startPlayAndCollectInteraction
Obtain call session
Play audio message
An audio message is played to the user
The user responds by entering a set of digits that are recorded by the service
Collect digits
notifyPlayAndCollectEvent
stopMediaInteractionNotification

Figure 4

6.5 Play media file and retrieve media from end user

This example shows an application playing a media file, retrieving media from the end user and recording it.

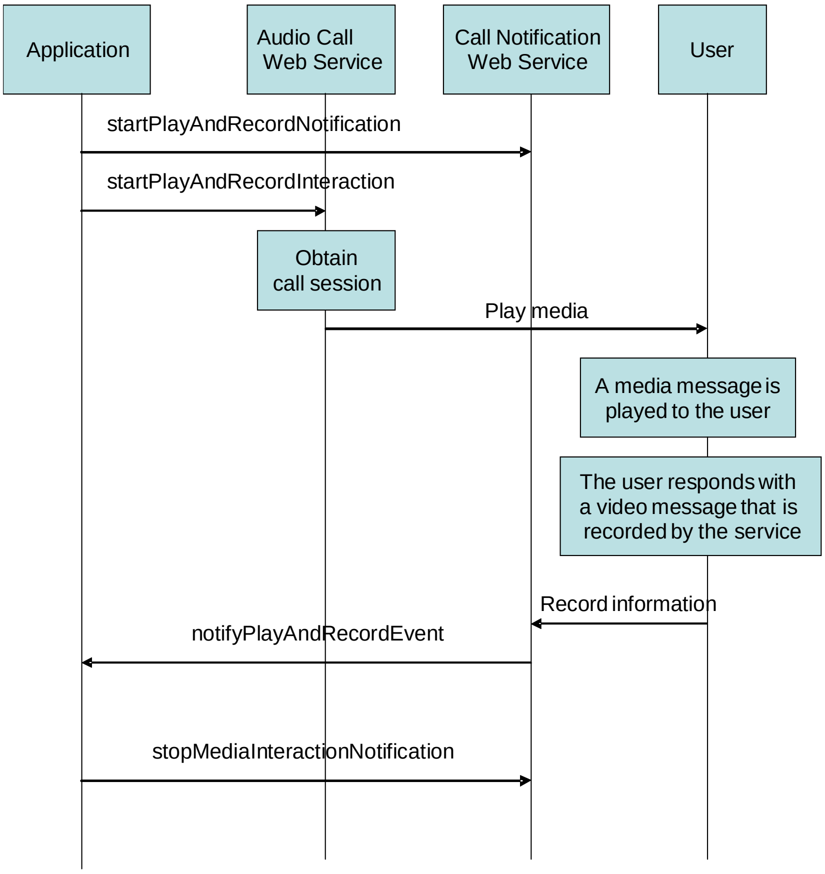
Application
Audio Call Web Service
Call Notification Web Service
User
startPlayAndRecordNotification
startPlayAndRecordInteraction
Obtain call session
Play media
A media message is played to the user
The user responds with a video message that is recorded by the service
Record information
notifyPlayAndRecordEvent
stopMediaInteractionNotification

Figure 5

7 XML Schema data type definition

7.1 MessageStatus enumeration

Status of the message after play message operation has been invoked.

Enumeration value	Description
Played	Message has been played
Playing	Message is currently playing
Pending	Message has not yet started playing
Error	An error has occurred, message will not be played

7.2 DigitConfig structure

Configuration parameters related to digit collection.

Element name	Element type	Optional	Description
maxDigits	xsd:int	Yes	The maximum number of digits that will be collected.
minDigits	xsd:int	Yes	The minimum number of digits that will be collected. If this isn't achieved, then a default prompt shall be played requesting that more digits be entered.
interruptMedia	xsd:boolean	No	Indicates whether the application allows the end user to interrupt, or pause, the playing of the prompt.

7.3 RecConfig structure

Configuration parameters related to media recording.

Element name	Element type	Optional	Description
recFileLocation	xsd:anyURI	Yes	The location for storing the information recorded from the terminal
maxRecordingLength	common:TimeMetric	Yes	The maximum duration of the media recording

7.4 PlayConfig structure

Configuration parameters related to the playing of a media file.

Element name	Element type	Optional	Description
playFileLocation	xsd:anyURI	Yes	The location of the file that will be played to the endpoint, including VoiceXML script location
textString	xsd:string	Yes	The text to be converted by a Text-To-Speech engine
messageFormat	AnnouncementFormat	No	The type of announcement prompt to play to the end user
interruptMedia	xsd:boolean	No	Indicates whether the application allows the end user to interrupt, or pause, the playing of the prompt.

7.5 AnnouncementFormat enumeration

The type of announcement prompt to play to an end user.

Enumeration value	Description
Audio	Announcement is in Audio format
VoiceXML	Announcement is in VoiceXML format
TextToSpeech	Announcement is in TextToSpeech format
Video	Announcement is in Video format
ApplicationSpecificFormat	Announcement is in an ApplicationSpecificFormat

7.6 MediaMessageStatus structure

Playing status of the message for each call participant after message operation has been invoked.

Element name	Element type	Optional	Description
callParticipant	xsd:anyURI	No	Participant address associated with correlator
status	MessageStatus	No	Current playing status for the participant

7.7 MediaParticipantInfo structure

Element name	Element type	Optional	Description
callParticipant	xsd:anyURI	No	Call Participant identifier
mediaInfo	common:MediaInfo [1..unbounded]	No	Information about media currently used by a call participant

8 Web Service interface definition

8.1 Interface: PlayMedia

The **PlayMedia** interface allows the playing of media messages using different forms of media content, and operations to monitor or cancel requests.

In all operations, a **callSessionIdentifier** is used to indicate the ongoing call session to which the request applies. If the call session is not valid, the Parlay X web service shall raise an exception: invalid input value. All operations may also list the set of **callParticipants** to which the requested media shall be provided. The set of participant addresses is restricted to a subset of the valid participant addresses associated with the identified call session.

8.1.1 Operation: playTextMessage

The invocation of **playTextMessage** requests play text identified by **text,** to the set of call participants within the call session specified by **callSessionIdentifier**. The text will be read through a Text-to-Speech engine, according to the specified **language**. The invocation returns as soon as the request is received by the system, i.e. the actual call is performed asynchronously. The **correlator**, returned by the invocation, can be used to identify the request, e.g. to get information on the request status.

This operation is intended to play a message to each of the specified call participants. If no call addresses are specified, the Parlay X Web Service shall play the message to all call participants within the call session specified by **callSessionIdentifier**. If call participants are specified, then only those call participants specified within the call session shall be played the message. The latter is to allow a message to be played to just one (or some) of the participants, where not all the participants in the call session are to receive the message.

All occurrences of invalid **callSessionIdentifier** or **callParticipants** shall result in an invalid input value exception.

8.1.1.1 Input message: playTextMessageRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Identifies the call session to which the message shall be played
callParticipants	xsd:anyURI [0..unbounded]	Yes	The set of participant addresses, within the call session specified by **callSessionIdentifier**, to which the message is to be played
text	xsd:string	No	Text to process with a Text-To-Speech engine
language	xsd:string	No	Language of text (ISO string)
charging	common:Charging Information	Yes	Charge to apply for the playing of this message. If charging is not supported then a PolicyException (POL0008) will be returned

8.1.1.2 Output message: playTextMessageResponse

Part name	Part type	Optional	Description
result	xsd:string	No	Correlator for this message for subsequent interactions

8.1.1.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0002: Privacy error.
- POL0006: Groups not allowed.
- POL0008: Charging not supported.

8.1.2 Operation: playAudioMessage

The invocation of **playAudioMessage** requests to play an audio file located at **audioUrl,** to the set of call participants within the call session specified by **callSessionIdentifier**. The invocation returns as soon as the request is received by the system, i.e. the actual call is performed asynchronously. The **correlator**, returned by the invocation, can be used to identify the request, e.g. to get information on the request status.

This operation is intended to play a message to each of the specified call participants. If no call addresses are specified, the Parlay X Web Service shall play the message to all call participants within the call session specified by **callSessionIdentifier**. If call participants are specified, then only those call participants specified within the call session shall be played the message. The latter is to allow a message to be played to just one (or some) of the participants, where not all the participants in the call session are to receive the message.

All occurrences of invalid **callSessionIdentifier** or **callParticipants** shall result in an invalid input value exception.

8.1.2.1 Input message: playAudioMessageRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Identifies the call session to which the message shall be played
callParticipants	xsd:anyURI [0..unbounded]	Yes	The set of participant addresses, within the call session specified by **callSessionIdentifier**, to which the message is to be played
audioUrl	xsd:anyURI	No	Location of audio content to play
charging	common:Charging Information	Yes	Charge to apply for the playing of this message. If charging is not supported then a PolicyException (POL0008) will be returned

8.1.2.2 Output message: playAudioMessageResponse

Part name	Part type	Optional	Description
result	xsd:string	No	Correlator for this message for subsequent interactions

8.1.2.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

- POL0002: Privacy error.
- POL0006: Groups not allowed.
- POL0008: Charging not supported.

8.1.3 Operation: playVoiceXmlMessage

The invocation of **playVoiceXmlMessage** requests to process VoiceXML content located at **voiceXmlUrl,** to the set of call participants within the call session specified by **callSessionIdentifier**. The invocation returns as soon as the request is received by the system, i.e. the actual call is performed asynchronously. The **correlator**, returned by the invocation, can be used to identify the request, e.g. to get information on the request status.

This operation is intended to play a message to each of the specified call participants. If no call addresses are specified, the Parlay X Web Service shall play the message to all call participants within the call session specified by **callSessionIdentifier**. If call participants are specified, then only those call participants specified within the call session shall be played the message. The latter is to allow a message to be played to just one (or some) of the participants, where not all the participants in the call session are to receive the message.

All occurrences of invalid **callSessionIdentifier** or **callParticipants** shall result in an invalid input value exception.

8.1.3.1 Input message: playVoiceXmlMessageRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Identifies the call session to which the message shall be played
callParticipants	xsd:anyURI [0..unbounded]	Yes	The set of participant addresses, within the call session specified by **callSessionIdentifier**, to which the message is to be played
voiceXmlUrl	xsd:anyURI	No	Location of VoiceXML content to process
charging	common:Charging Information	Yes	Charge to apply for the playing of this message. If charging is not supported then a PolicyException (POL0008) will be returned

8.1.3.2 Output message: playVoiceXMLMessageResponse

Part name	Part type	Optional	Description
result	xsd:string	No	Correlator for this message for subsequent interactions

8.1.3.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0002: Privacy error.
- POL0006: Groups not allowed.
- POL0008: Charging not supported.

8.1.3a Operation: playVideoMessage

The invocation of **playVideoMessage** requests to play a video stream identified by **video,** to the set of call participants within the call session specified by **callSessionIdentifier**. The invocation returns as soon as the request is received by the system, i.e. the actual call is performed asynchronously. The **correlator**, returned by the invocation, can be used to identify the request, e.g. to get information on the request status.

This operation is intended to play video to each of the specified call participants. If no call addresses are specified, the Parlay X Web Service shall play the video to all call participants within the call session specified by **callSessionIdentifier**. If call participants are specified, then only those call participants specified within the call session shall be played the video. The latter is to allow a video to be played to just one (or some) of the participants, where not all the participants in the call session are to receive the video.

All occurrences of invalid **callSessionIdentifier** or **callParticipants** shall result in an invalid input value exception.

8.1.3a.1 Input message: playVideoMessageRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Identifies the call session to which the message shall be played
callParticipants	xsd:anyURI [0..unbounded]	Yes	The set of participant addresses, within the call session specified by **callSessionIdentifier**, to which the message is to be played
video	xsd:anyURI	No	Identifies the video content to process
charging	common:Charging Information	Yes	Charge to apply for the playing of this message. If charging is not supported then a PolicyException (POL0008) will be returned

8.1.3a.2 Output message: playVideoMessageResponse

Part name	Part type	Optional	Description
result	xsd:string	No	Correlator for this message for subsequent interactions

8.1.3a.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0002: Privacy error.
- POL0006: Groups not allowed.
- POL0008: Charging not supported.

8.1.4 Operation: getMessageStatus

The invocation of **getMessageStatus** retrieves the current status, **result**, of a previous request identified by **correlator**.

8.1.4.1 Input message: getMessageStatusRequest

Part name	Part type	Optional	Description
correlator	xsd:string	No	Correlator returned from play operation to check

8.1.4.2 Output message: getMessageStatusResponse

Part name	Part type	Optional	Description
result	MediaMessageStatus [1..unbounded]	No	Current playing status for each call participant related to correlator

8.1.4.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.1.5 Operation: endMessage

The invocation of **endMessage** cancels/stops a previous request identified by **correlator**. It returns a **result**, with the status of the request at the moment of abort.

8.1.5.1 Input message: endMessageRequest

Part name	Part type	Optional	Description
correlator	xsd:string	No	Correlator returned from play operation to cancel

8.1.5.2 Output message: endMessageResponse

Part name	Part type	Optional	Description
result	MediaMessageStatus [1..unbounded]	No	Status of message operation, for each call participant related to correlator, at the time the endMessage was acted on

8.1.5.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.2 Interface: CaptureMedia

8.2.1 Operation: startPlayAndCollectInteraction

The application shall invoke this operation in order to play a media file to either one or all call parties on an existing call and collect digits from a call party. The call shall be identified using the **callSessionIdentifier** part of the request message. If the **callParticipant** part is provided, then the media interaction is limited to this participant on the call as opposed to the entire call.

The **playingConfiguration** part shall contain all the information about the announcement to be played to the participant or call. The **digitConfiguration** part shall contain the configuration parameters for the digit collection.

The response message shall contain a media identifier that can be used by the application, if so desired, to interrupt an ongoing media interaction using the **stopMediaInteraction** operation.

8.2.1.1 Input message: startPlayAndCollectInteractionRequest

Part Name	Part Type	Optional	Description
callSessionIdentifier	xsd:string	No	Identifies the call session for the media interaction.
callParticipant	xsd:anyURI	Yes	If this is present, the media interaction is with this call participant only. If this is not present, the media interaction is with all participants on the call
playingConfiguration	PlayConfig	No	Configuration parameters related to the playing of a media file
digitConfiguration	DigitConfig	No	Configuration parameters related to digit collection

8.2.1.2 Output message: startPlayAndCollectInteractionResponse

Part Name	Part Type	Optional	Description
result	xsd:string	No	An identifier that uniquely defines the media interaction

8.2.1.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0002: Privacy error

8.2.2 Operation: startPlayAndRecordInteraction

The application shall invoke this operation in order to play a media file to either one or all call parties on an existing call and record information (media) from a call party. The call shall be identified using the **callSessionIdentifier** part of the request message. If the **callParticipant** part is provided, then the media interaction is limited to this participant on the call as opposed to the entire call.

The **playingConfiguration** part contains all the information about the announcement to be played to the participant or call. The **recordingConfiguration** part shall contain the configuration parameters for the media recording.

The response message shall contain a media identifier that can be used by the application, if so desired, to interrupt an ongoing media interaction using the **stopMediaInteraction** operation.

8.2.2.1 Input message: startPlayAndRecordInteractionRequest

Part Name	Part Type	Optional	Description
callSessionIdentifier	xsd:string	No	Identifies the call session for the media interaction.
callParticipant	xsd:anyURI	Yes	If this is present, the media interaction is with this call participant only. If this is not present, the media interaction is with all participants on the call
playingConfiguration	PlayConfig	No	Configuration parameters related to the playing of a media file
recordingConfiguration	RecConfig	No	Configuration parameters related to media recording

8.2.2.2 Output message: startPlayAndRecordInteractionResponse

Part Name	Part Type	Optional	Description
result	xsd:string	No	An identifier that uniquely defines the media interaction

8.2.2.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0002: Privacy error

8.2.3 Operation: stopMediaInteraction

This operation shall stop an ongoing media interaction. The **mediaIdentifier** part provided by the application in the request message shall contain the value returned in the response message of the associated **startPlayAndCollectInteraction** or **startPlayAndRecordInteraction** operation.

The **stopMediaInteraction** operation is only required in order to interrupt an ongoing interaction such as on hold music. Many interactions have a natural endpoint (e.g. collecting digits) and in this case the **stopMediaInteraction** is not required.

8.2.3.1 Input message: stopMediaInteractionRequest

Part Name	Part Type	Optional	Description
mediaIdentifier	xsd:string	No	An identifier that uniquely defines the media interaction

8.2.3.2 Output message: stopMediaInteractionResponse

Part Name	Part Type	Optional	Description
None			

8.2.3.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.3 Interface: Multimedia

The **Multimedia** interface can be used by an application for dynamically managing the media types for the participants involved in the call.

8.3.1 Operation: addMediaForParticipants

The application invokes the **addMediaForParticipants** operation to add **media** type(s) to the media set used by all of the call participants identified. If no call participant addresses are specified, the Audio Call Web Service shall request addition of the media type(s) specified to all call participants in the call session (i.e. associated with the **callSessionIdentifier)**. If call participants are specified, then only those call participants explicitly specified shall have the media type(s) added. The latter is to allow media type(s) to be added or changed for just one (or some) of the participants, where not all the participants in the call session are to have exactly the same set of media types.

The added media type has to be compatible with the set of media types supported by the participant's device, otherwise the operation will fail. The resultant media details can be retrieved using **getMediaForParticipant** or **getMediaForCall.** If a participant has already the requested media type and direction, the operation shall fail and throw ServiceException SVC0290.

8.3.1.1 Input message: addMediaForParticipantsRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Call session identifier. It identifies the existing call session or call conference. This must be a non-null value as it identifies a pre-existing call (or conference) in the network
callParticipants	xsd:anyURI [0..unbounded]	Yes	Call participant(s). The set of participant addresses contained within the call session to which the add media stream is to apply.
mediaInfo	common:MediaInfo [1..unbounded]	No	It identifies the media type(s) the participant(s) requested to be able to receive/send and the desired direction of the media stream(s), i.e. incoming, outgoing or bidirectional. At least one media type shall be specified.

8.3.1.2 Output message: addMediaForParticipantsResponse

Part name	Part type	Optional	Description
None			

8.3.1.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0290: Duplicate media type

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0011: Media type not supported

8.3.2 Operation: deleteMediaForParticipants

The application invokes the **deleteMediaForParticipants** operation to delete **media** types from the media set used by all the call participants identified. If no call participant addresses are specified, the requested media type(s) for all call participants in the call session (i.e. associated with the **callSessionIdentifier)** shall be removed. If call participants are specified, then only those call participants explicitly specified shall have the media type(s) removed. The latter is to allow media type(s) to be removed for just one (or some) of the call participants, where not all the call participants in the call are to have exactly the same set of media types. The resultant media details can be retrieved using the **getMedia ForParticipant** or **getMediaForCall** operations.

If a call participant does not have the requested media type, the operation shall fail and throw ServiceException SVC0291.

8.3.2.1 Input message: deleteMediaForParticipantsRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Call session identifier. It identifies the existing call session or call conference. This must be a non-null value as it identifies a pre-existing call (or conference) in the network
callParticipants	xsd:anyURI [0..unbounded]	Yes	Call participant(s). The set of participant addresses contained within the call session to which the delete media type(s) is to apply.
media	common:Media [1..unbounded]	No	It identifies the media type(s) that are not to be used any more by the participant(s). At least one media type shall be specified.

8.3.2.2 Output message: deleteMediaForParticipantsResponse

Part name	Part type	Optional	Description
None			

8.3.2.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0291: Media stream does not match

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.3.3 Operation: getMediaForParticipant

The application invokes the **getMediaForParticipant** operation to request information concerning the current media status of a single participant of the multi-media call identified by **callSessionIdentifier**.

8.3.3.1 Input message: getMediaForParticipantRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Call session identifier. It identifies the existing call session or call conference. This must be a non-null value as it identifies a pre-existing call (or conference) in the network
callParticipant	xsd:anyURI	No	Identifies a specific call participant within the call session

8.3.3.2 Output message: getMediaForParticipantResponse

Part name	Part type	Optional	Description
result	common:MediaInfo [1..unbounded]	No	Array containing media status information for the requested call participant

8.3.3.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.3.4 Operation: getMediaForCall

The application invokes the **getMediaForCall** operation to request information concerning the current media status of each participant of the multi-media call identified by **callSessionIdentifier.**

8.3.4.1 Input message: getMediaForCallRequest

Part name	Part type	Optional	Description
callSession Identifier	xsd:string	No	Call session identifier. It identifies the existing call session or call conference. This must be a non-null value as it identifies a pre-existing call (or conference) in the network

8.3.4.2 Output message: getMediaForCallResponse

Part name	Part type	Optional	Description
result	MediaParticipantInfo [1..unbounded]	No	Array containing media status information for each call participant

8.3.4.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

9 Fault definitions

9.1 ServiceException

9.1.1 SVC0290: Duplicate media stream

Name	Description
messageId	SVC0290
text	Duplicate media stream
variables	None

9.1.2 SVC0291: Media stream does not match

Name	Description
messageId	SVC0291
text	Media stream does not match type specified
variables	None

10 Service policies

Service policies for this service.

Name	Type	Description
TextToSpeechAvailable	xsd:boolean	Indicates whether the service accepts text as an input for processing with a Text-To-Speech engine
AudioContentAvailable	xsd:boolean	Indicates whether the service accepts audio content for playing with an audio player
VoiceXMLAvailable	xsd:boolean	Indicates whether the service accepts VoiceXML as an input for processing with a VoiceXML browser
DigitCollectionAvailable	xsd:boolean	Service accepts digit collection input from the end user
RecordMessageAvailable	xsd:boolean	Service accepts recorded message input from the end user
StatusRetensionTime	common:TimeMetric	Time interval for which status is retained after a message is played or an error occurs
AudioFormatsSupported	xsd:string	Comma separated list of audio formats supported (e.g. WAV, MP3, AU)
ChargingSupported	xsd:boolean	Indicates whether charging is supported for the play operations
VideoAvailable	xsd:boolean	Indicates whether the service accepts video content for streaming.
MultimediaSupported	xsd:boolean	Indicates whether multimedia is supported and whether an application can change the media types used in a call.
MinDigits	xsd:int	The minimum number of collected digits supported for interaction with the end-user
MaxDigits	xsd:int	The maximum number of collected digits supported for interaction with the end-user
MaxRecordingLength	common:TimeMetric	Time interval indicating the maximum length of time for which end-user input will be recorded.

Annex A (normative): WSDL for Audio Call

The document/literal WSDL representation of this interface specification is compliant to ES 202 504-1 [2] and is contained in text files (contained in archive es_20250411v010101p0.zip) which accompany the present document.

Annex B (informative): Bibliography

ETSI TR 121 905: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

History

Document history		
V1.1.1	February 2008	Membership Approval Procedure MV 20080425: 2008-02-26 to 2008-04-25
V1.1.1	May 2008	Publication