ETSI ES 202 504-19 V1.1.1 (2008-05)

ETSI Standard
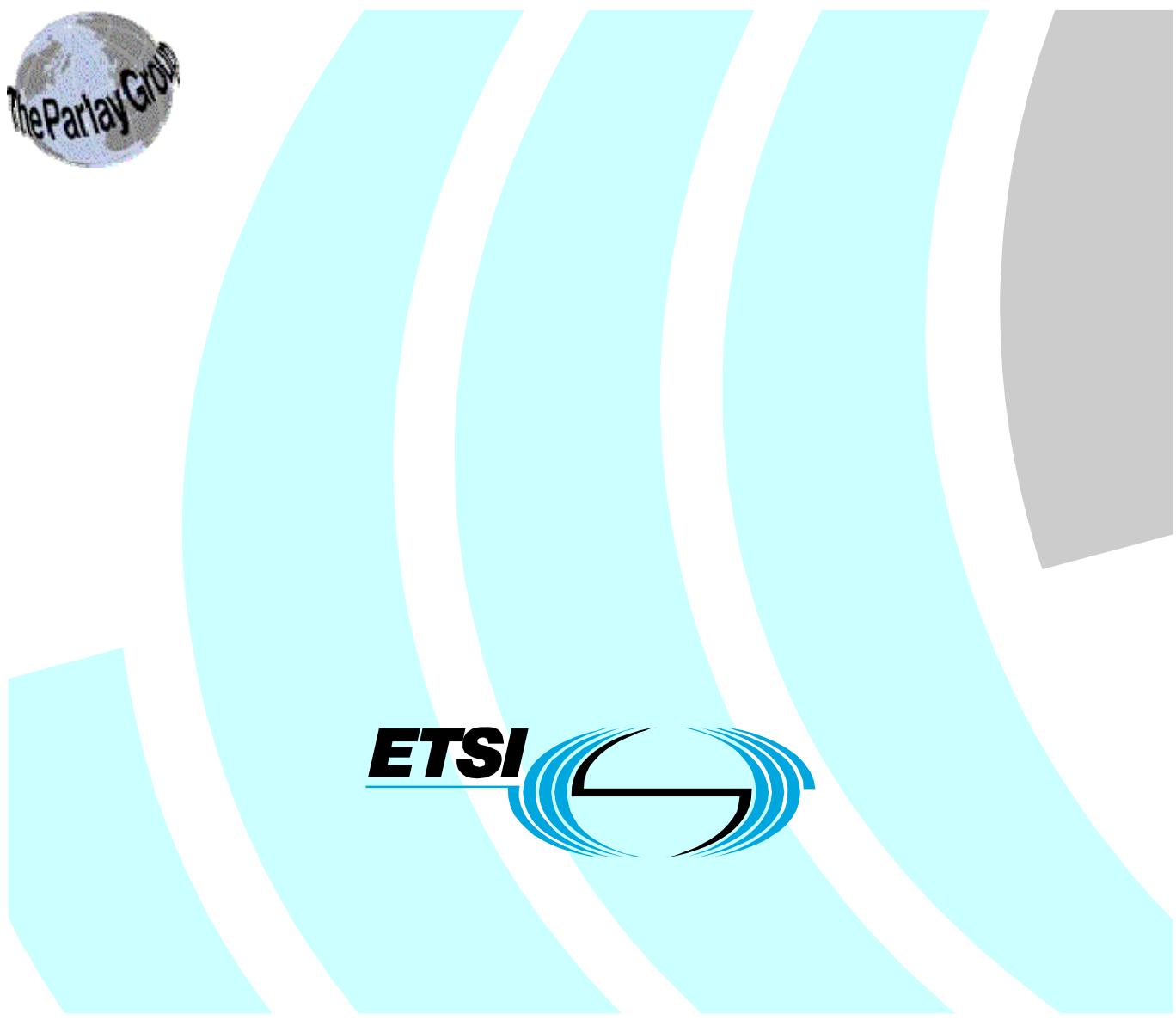
**Open Service Access (OSA);
Parlay X Web Services;
Part 19: Multimedia Streaming Control
(Parlay X 3)**

Reference

DES/TISPAN-01034-19-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

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or
perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF).
In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive
within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

ETSI

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 19 of a multi-part deliverable covering Open Service Access (OSA); Parlay X 3 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 29.199-19 V7.0.2 (Release 7).

1 Scope

The present document is part 19 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 Multimedia Streaming Control 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.

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]   ETSI ES 202 504-6: "Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (Parlay X 3)".

[4]   ETSI ES 204 915-7: "Open Service Access (OSA); Application Programming Interface (API); Part 7: Terminal Capabilities SCF (Parlay 6)".

[5]   ISO 4217: "Codes for the representation of currencies and funds".

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 defined in ES 202 504-1 [2] apply.

4   Detailed service description

The service provided to an end-user is consumption of streaming multimedia. The end-user has a terminal that is able to request a media stream, either from a built-in player, or an installed application. The terminal can be any terminal with streaming media playing capabilities, and the service should allow a user to transfer between his terminals.

- The basic scenario is where an individual is browsing the Internet and finds some interesting content that he/she wants to watch. The end-user is then either doing this through the operator's portal or accesses the content provider's site. In the first case, the request is then processed through the portal, and charged as the stream is set-up. In the second case, the content provider redirects the request to the operator, so that the terminal capabilities can be collected and charging is done, before the stream is started.

- One scenario is an individual watching his favourite sports stream on his TV or PC at home, but he/she must leave the house of some reason, and still wants to continue the session on his mobile terminal. In that case he/she would transfer the ongoing session to the other terminal with other capabilities, since he already paid for the entire episode.

- The other way is a similar case where an individual arrives at a destination with better viewing capabilities.

4.1   Overview

The figure below shows streaming content delivered to mobile terminals, but is not restricted to any particular terminal type. The content is either on a location accessible over the Internet or stored locally on an operator's domain or the content provider's domain. The access to the content is done through a smart router controlling the stream towards the terminal. Transcoding of a stream is an optional feature that could allow users to switch between different terminal and networks, while consuming content from a streaming source.
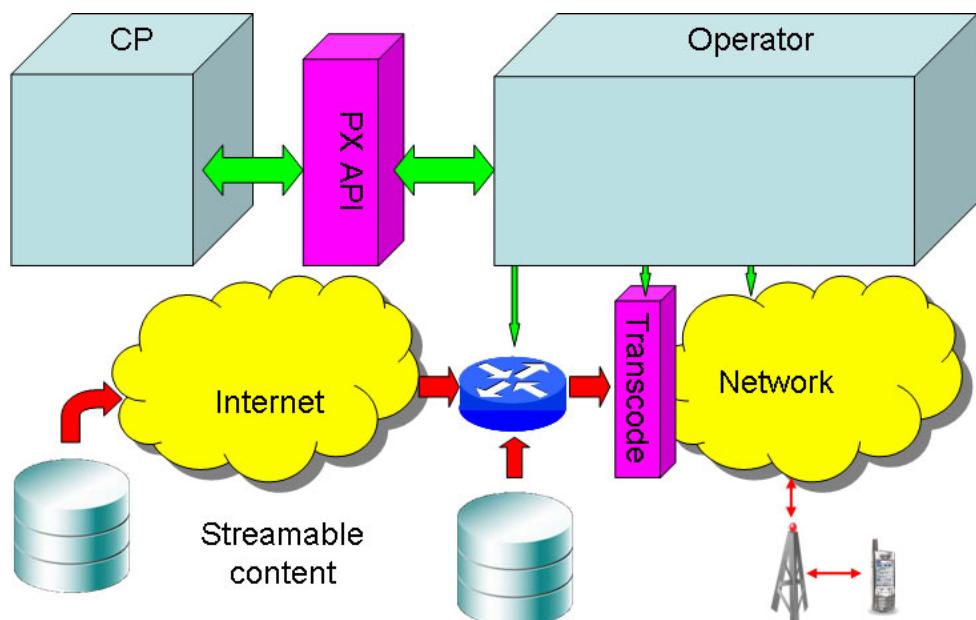
Figure 4.1.1

5   Namespaces

The Streaming interface uses the namespace:

   http://www.csapi.org/wsdl/parlayx/streaming/ v3_0

The StreamingNotificationManager interface uses the namespace:

   http://www.csapi.org/wsdl/parlayx/ streaming /notification_manager/v3_0

The StreamingNotification interface uses the namespace:

   http://www.csapi.org/wsdl/parlayx/ streaming /notification/v3_0

The data types are defined in the namespace:

   http://www.csapi.org/schema/parlayx/ streaming /v3_0

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

The following use case diagram defines the core functionality which should be supported by the Multimedia Streaming
Control web service. These high level logical use cases will be supported by the defined API, but not necessarily on a
one to one mapping basis.

StreamControl Web Service

- Request Media Stream
- Start Media Stream
- Redirect Media Stream
- Pause Media Stream
- Resume Media Stream
- Terminate Media Stream
- Modify Media Stream
- Request Session Activity Status
- Refund End User Charges
- Subscribe Notification Events
- Notify Stream Started
- Notify Stream Completed
- Notify Terminal Status
- Notify Stream Interrupted / Cancelled

ContentProvider

Operator

Charging Information contained in request

May includes consumer terminal data - e.g. connection type, capabilities (e.g. codecs supported)

Figure 6.1

6.1   Getting requests

The following diagram illustrates the scenario where incoming stream requests are buffered at the operator. The content provider periodically retrieves streaming requests and handles them.
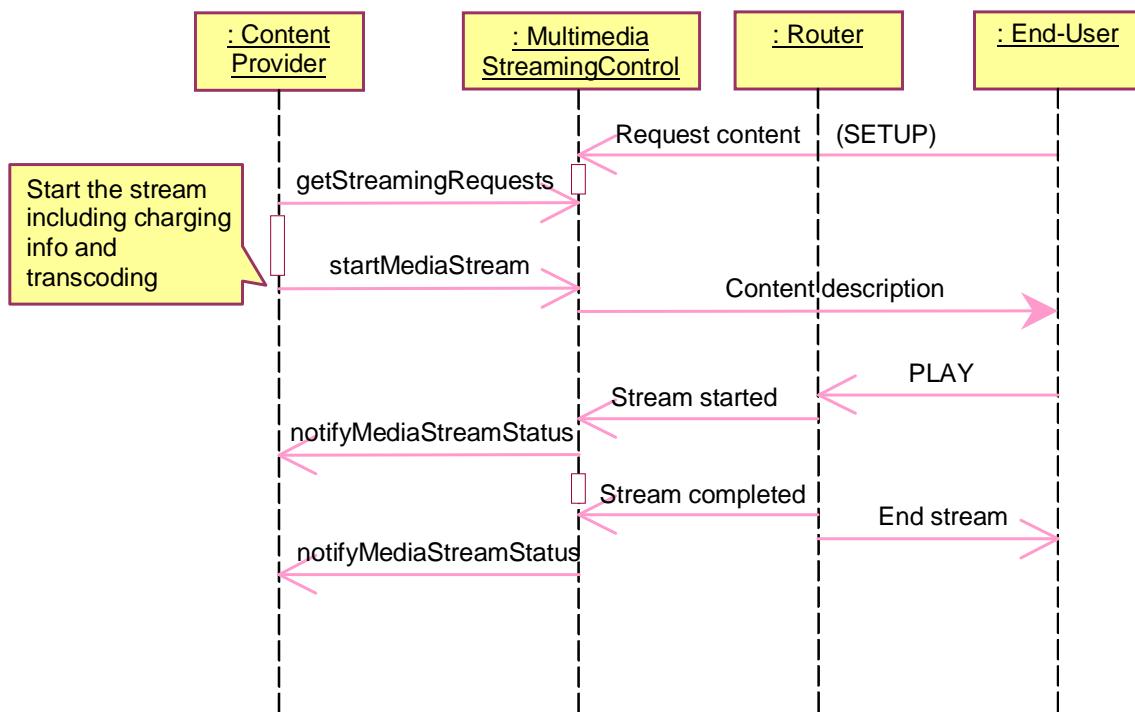
6.2   Charging refund

The following diagram illustrates the scenario where incoming stream requests are sent to the content provider when received by the operator. In this scenario the stream is cancelled before playing is completed and the content provider refunds an amount previously requested.
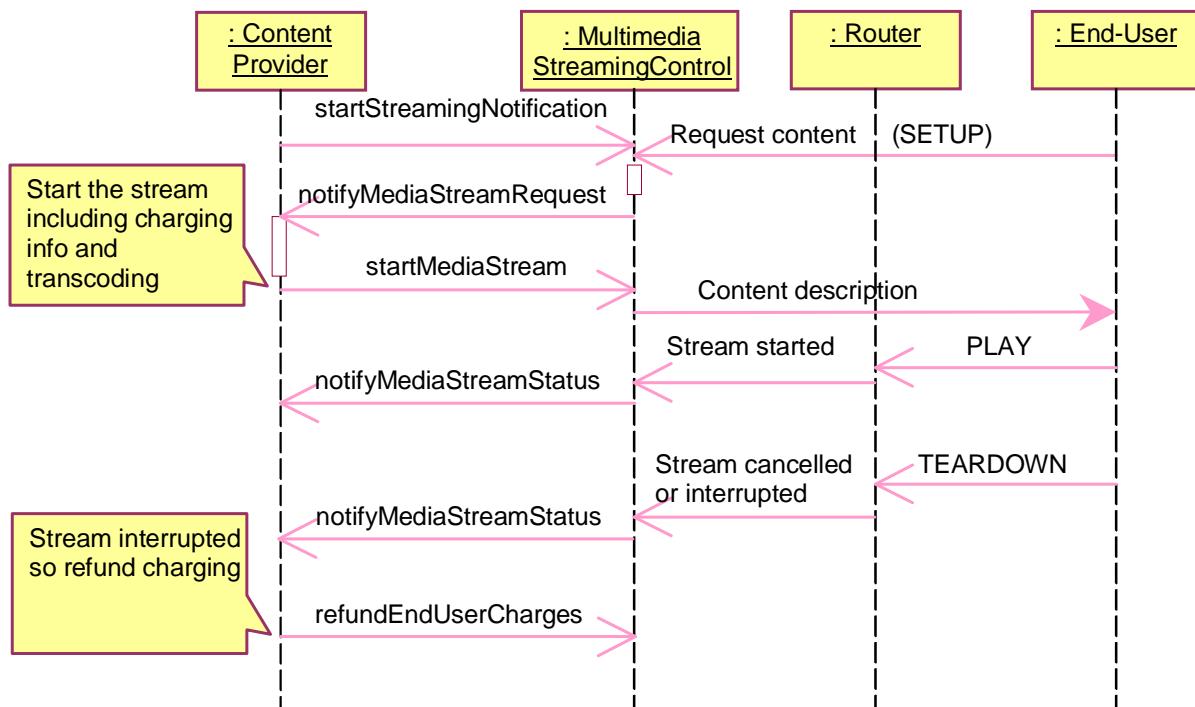
6.3 Trans-coding update

The following diagram illustrates the scenario where the capabilities for a terminal or a network changes, for example due to handover between GPRS and 3G. The contents provider is notified about this change and can change for example transcoding of the stream.
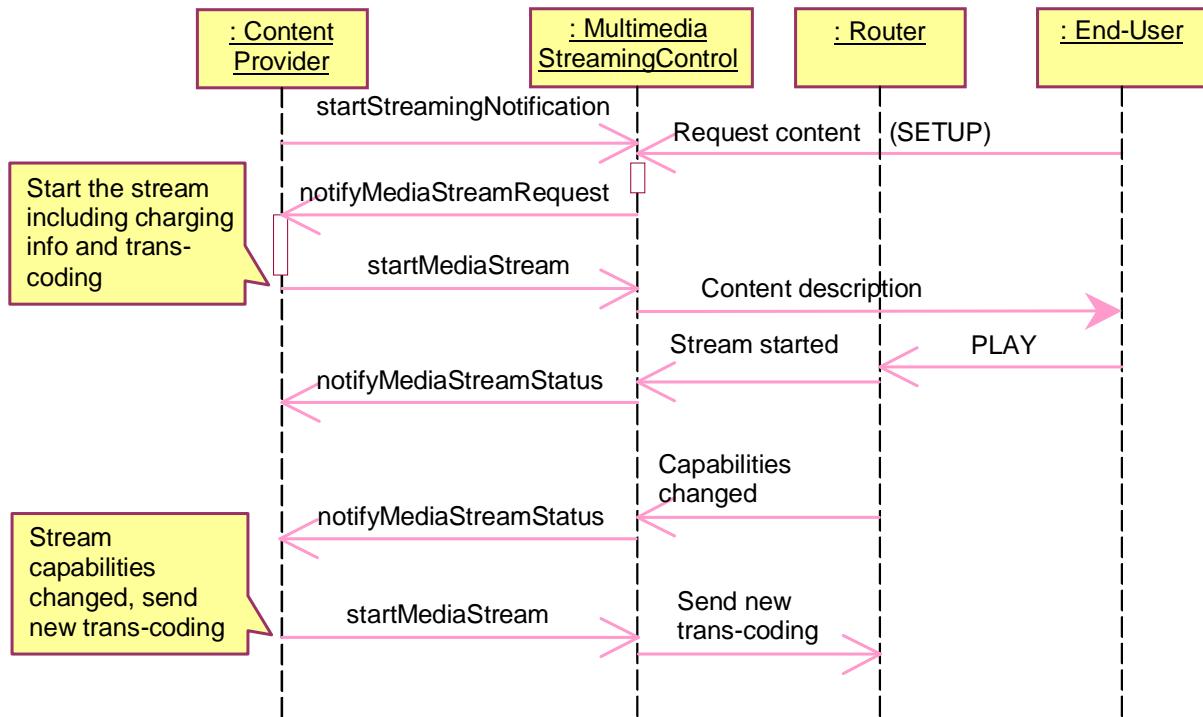
6.4 Controlling multiple streams

The following diagram illustrates the scenario where the content provider pauses an ongoing stream and temporarily plays another stream, for example an advertisement. When the second stream ends the first original stream is continued. From the end-user's point of view this should be perceived as one uninterrupted stream.

The following sequence diagram shows the following flow:

: Content Provider — **: Multimedia StreamingControl** — **: Router** — **: End-User**

- startStreamingNotification (Content Provider → Multimedia StreamingControl)
- Request content (SETUP) (End-User → Multimedia StreamingControl)

Start the stream including charging info and transcoding

- notifyMediaStreamRequest (Multimedia StreamingControl → Content Provider)
- startMediaStream (Content Provider → Multimedia StreamingControl)
- Content description (Multimedia StreamingControl → End-User)
- Stream 1 started / PLAY (End-User → Router)

Content provider pauses stream and plays an other stream

- notifyMediaStreamStatus (Router → Content Provider)
- controlMediaStream (Content Provider → Multimedia StreamingControl)
- Pause stream 1 (Multimedia StreamingControl → Router)
- Stream 2 started (Router → Multimedia StreamingControl)
- Stream 2 finished (Router → Multimedia StreamingControl)

Second stream finished, resume the first stream

- notifyMediaStreamStatus (Multimedia StreamingControl → Content Provider)
- controlMediaStream (Content Provider → Multimedia StreamingControl)
- Resume stream 1 (Multimedia StreamingControl → Router)
- Stream 1 resumed (Router → Multimedia StreamingControl)
- Stream 1 finished (Router → Multimedia StreamingControl)
- notifyMediaStreamStatus (Multimedia StreamingControl → Content Provider)
- End stream (Router → End-User)

6.5   Redirecting a stream

The following diagram illustrates the scenario where a stream is being redirected to another device due to a user request. The contents provider receives the request from the end-user and continues the stream to the new device.
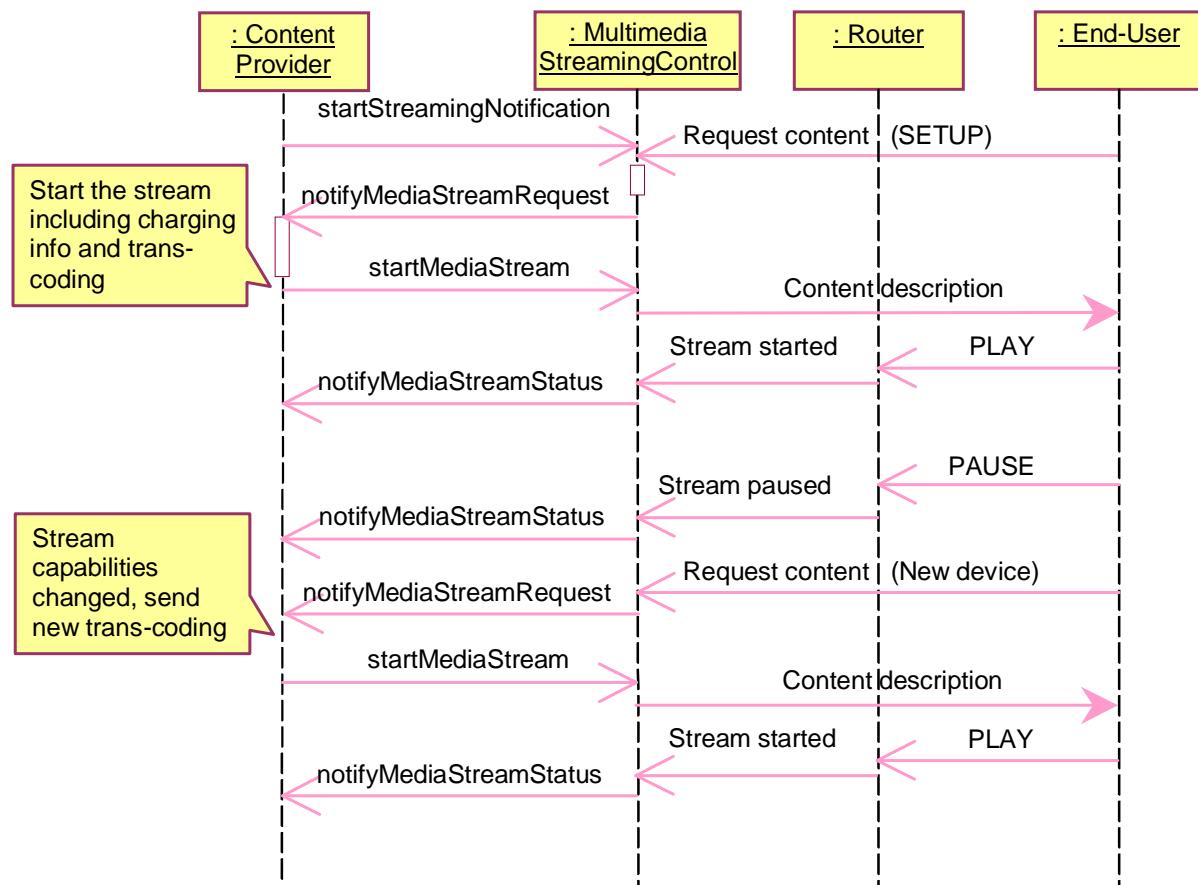
6.6   Multimedia Stream Control with Dynamic QoS

The following sequence diagram provides an example of how an application could combine functionality from the Multimedia Streaming Control web service (this specification) with the Application Driven QoS (ADQ) web service to provide guaranteed media stream delivery.

Flow summary:

1.   Content Provider receives a NotifyMediaStreamRequest which includes the content identifier, terminal capabilities and an estimated available bandwidth of the current network session.

2.   Based upon content requirements and current network bandwidth, the content provider may decide to provide a guaranteed QoS for content delivery by applying a temporary QoS session (addTemporaryQoSFeature on ADQ Web Service).

3.   The content provider will then continue to setup the media stream through the StartMediaStream operation, as for previous scenarios.

4.   At some point during the ongoing stream, the network / terminal capabilities change and the content provider is notified by the operator with the NotifySessionStatus operation.

5.   Based on the new session parameters (e.g. available bandwidth), the content provider may modify the guaranteed QoS parameters previously requested (modifyTemporaryQoSFeature).

6.   Upon stream completion, content provider will remove the previously requested QoS session (removeTemporaryQosFeature).
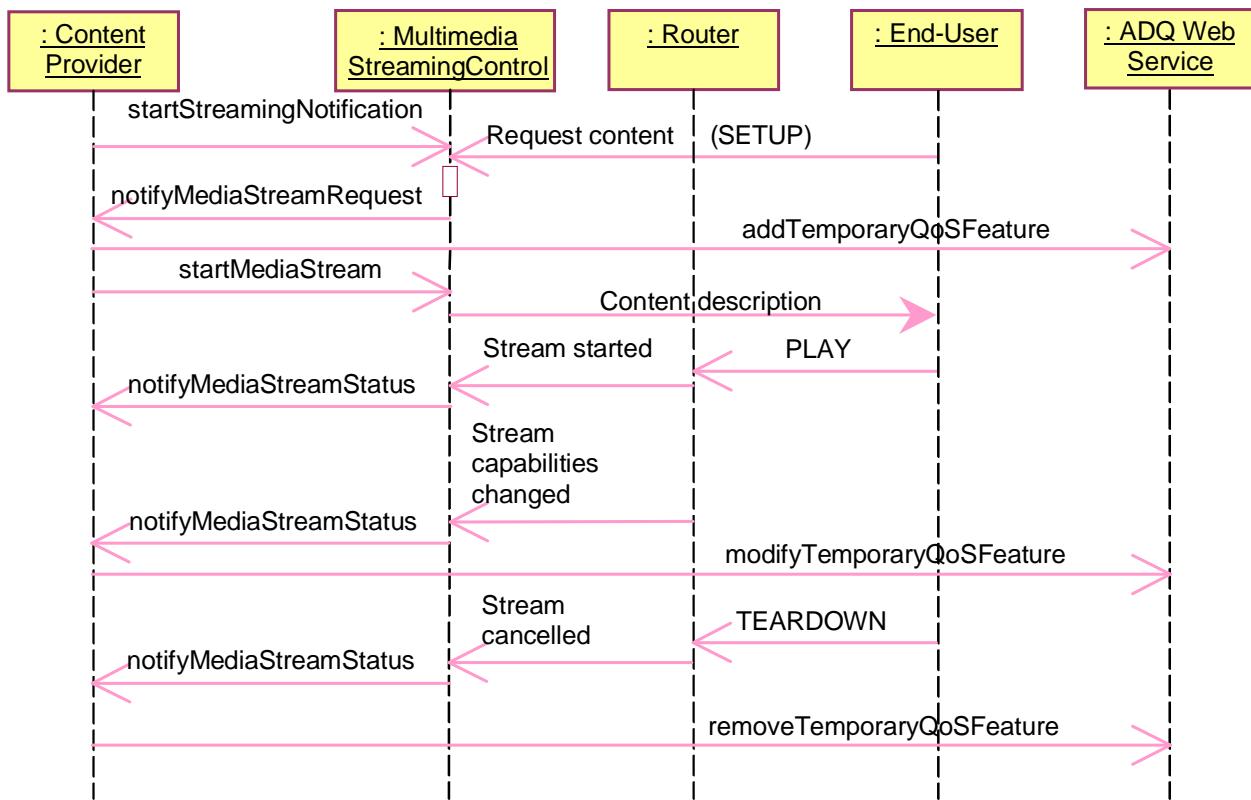
7 XML Schema data type definition

7.1 MediaStreamRequest structure

Data type used to capture attributes of a single media stream request.

Element Name	Element Type	Optional	Description
subscriberIdentifier	xsd:anyURI	No	Identity of subscriber / end-user making the request for content streaming
contentIdentifier	xsd:anyURI	No	Identifier / Address of requested content item
sessionIdentifier	xsd:string	No	Operator provided session identifier
streamTransfer	xsd:boolean	Yes	States whether this request is a result of a stream transfer (e.g. transfer of existing stream from mobile device to a PC) or is a new media stream request.
terminalCapabilities	xsd:string [0..unbounded]	Yes	Terminal capability description derived from *TpTerminalCapabilities* (ref ES 204 915-7 [4])
availableBandwidth	xsd:int	Yes	Estimation of current available network bandwidth in bits per second

7.2 MediaStreamInformation structure

Composite data type required by the **getMediaStreamStatus** operation in order to return all requested information in a single message part..

Element Name	Element Type	Optional	Description
status	MediaStreamStatus	No	Communicates status of ongoing media stream
information	xsd:string	Yes	Contains data specific to the **status** parameter.

7.3 StreamControlAction enumeration

Set of enumerated actions which may be applied to an ongoing multimedia stream.

Enumeration value	Description
Pause	Pause ongoing stream
Resume	Resume previously paused stream
Terminate	Terminate an active media stream
Start	Start a stream
Stop	Stop a stream
Forward	Move forward in time on a stream
Rewind	Move backwards in time on a stream

7.4 MediaStreamStatus enumeration

Extensible list of Media Stream Status

Enumeration value	Description
Started	Media stream started
Ready	Media stream is ready to be played at router
Playing	Media stream is playing
Completed	Media stream completed.
Cancelled	Media stream cancelled by user
Interrupted	Media stream interrupted. E.g. loss of network signal coverage, terminal loses power etc.
Failed	Media stream delivery failed. E.g. permanent failure in delivery, no route to terminal etc.

8 Web Service interface definition

8.1 Interface: Streaming

8.1.1 Operation: startMediaStream

The **startMediaStream** operation is sent to the Parlay X server to specify how a stream should be handled by the operator. The information included is media server location, charging data and information about how to handle transcoding. The operation is sent as a response to either a previous received **notifyMediaStreamRequest** operation or to streaming requests received in the response to a **getStreamingRequests** operation.

8.1.1.1 Input message: startMediaStreamRequest

Part Name	Part Type	Optional	Description
sessionIdentifier	xsd:string	No	Session identifier provided by the operator in the initial **notifyMediaStreamRequest** or **getStreamingRequests** operation
contentLocation	xsd:anyURI	No	Location (URL) of content item to be delivered in streaming session
chargingInformation	common:Charging Information	Yes	Optionally allows content provider to specify charging information to be applied to media stream
allowNetworkTranscoding	xsd:boolean	Yes	Allows a content provider to specify whether or not the network operator may apply further transcoding to the content media stream

8.1.1.2   Output message: startMediaStreamResponse

Part Name	Part Type	Optional	Description
result	MediaStreamStatus	No	Current status

8.1.1.3   Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.

- SVC0002: Invalid input value.

- SVC0007: Invalid charging information.

- SVC0270: Charge failed.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.1.2   Operation: getStreamingRequests

The invocation of **getStreamingRequests** retrieves all the content requests received that fulfil the criteria identified by
the **contentProviderIdentifier** part. The operation returns only the list of content requests received since the previous
invocation of the same operation, i.e. each time the operation is invoked the content requests returned are removed from
the server. Moreover, each content request will be automatically removed from the server after a maximum time
interval as defined by a service policy. This operation allows the content provider to poll for a list of pending streaming
requests rather than receive dynamic notification (**notifyMediaStreamRequest**). This operation returns a list of
pending **MediaStreamRequest** structures.

8.1.2.1   Input message: getStreamingRequestsRequest

Part Name	Part Type	Optional	Description
contentProviderIdentifier	xsd:string	No	Identifies content provider.

8.1.2.2   Output message: getStreamingRequestsResponse

Part Name	Part Type	Optional	Description
result	MediaStreamRequest[0..unbounded]	Yes	List of pending media stream requests for the specified Content Provider

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.

- POL0010: Retention time interval expired.

8.1.3   Operation: controlMediaStream

Allows content provider to control an ongoing multimedia streaming session, as identified by the **sessionIdentifier** message part. The control actions allowed are as defined in the **StreamControlAction** enumeration: e.g. pause, resume, terminate, etc.

8.1.3.1   Input message: controlMediaStreamRequest

Part Name	Part Type	Optional	Description
sessionIdentifier	xsd:string	No	Session identifier provided by the operator in the initial **notifyMediaStreamRequest** or **getStreamingRequests** operation
action	StreamControlAction	No	Specifies action to be applied to active media stream.

8.1.3.2   Output message: controlMediaStreamResponse

Part Name	Part Type	Optional	Description
result	MediaStreamStatus	No	Current status

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.

8.1.4   Operation: getMediaStreamStatus

Allows the content provider to request current status (e.g. whether a stream has started, completed etc) on a previously requested, using **startMediaStream**, multimedia stream, which is identified by the **sessionIdentifier** message part . Returns the current status (as defined by the **MediaStreamStatus** enumeration) and, if applicable, any associated status parameters, e.g. start time, completion time.

8.1.4.1   Input message: getMediaStreamStatusRequest

Part Name	Part Type	Optional	Description
sessionIdentifier	xsd:string	No	Session identifier provided by the operator in the initial **notifyMediaStreamRequest** or **getStreamingRequests** operation

8.1.4.2   Output message: getMediaStreamStatusResponse

Part Name	Part Type	Optional	Description
result	MediaStreamInformation	No	Communicates status of ongoing media stream and any status-specific information.

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: refundEndUserCharges

Allows content provider to request that the operator apply a refund of charges to an end-user account. For example, this could be as a result of a multimedia stream failing before completion due to loss of network connectivity etc. (This would have been notified to the content provider through invocation of either a **getMediaStreamStatus** operation or a **notifyMediaStreamStatus** callback operation). Content provider will specify the refund amount within the **chargingInformation** message part.

8.1.5.1 Input message: refundEndUserChargesRequest

Part Name	Part Type	Optional	Description
subscriberIdentifier	xsd:anyURI	No	Identity of subscriber / end user.
sessionIdentifier	xsd:string	No	Session identifier provided by the operator in the initial **notifyMediaStreamRequest** or **getStreamingRequests** operation
chargingInformation	common:ChargingInformation	No	Specifies refund amount to be applied by operator to end-user's account.
referenceCode	xsd:string	Yes	Textual information to uniquely identify the request, e.g. in case of disputes

8.1.5.2 Output message: refundEndUserChargesResponse

Part Name	Part Type	Optional	Description
None			

8.1.5.3 Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.

- SVC0002: Invalid input value.

- SVC0007: Invalid charging information.

- SVC0270: Charge failed.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.

8.2 Interface: StreamingNotificationManager

8.2.1 Operation: startNotification

The notification pattern with correlation is used in order to correlate the notification events with the request.

Initial event registration operation which allows content provider's application to subscribe to receive event notifications from the web service relating to newly requested (i.e. **notifyMediaStreamRequest**) and ongoing multimedia streams (e.g. **notifyMediaStreamStatus**). Content provider will provide a call-back address and correlator string (which may subsequently be used to cancel notifications) within the reference parameter.

Note that the **SimpleReference** structure contains the **correlator** string used in subsequent messages to the content provider's **StreamingNotification** interface.

8.2.1.1   Input message: startNotificationRequest

Part Name	Part Type	Optional	Description
reference	common: SimpleReference	No	Notification endpoint definition
frequency	common: TimeMetric	No	Maximum frequency of notifications (can also be considered minimum time between notifications)
duration	common: TimeMetric	Yes	Length of time notifications occur for; do not specify to use default notification time defined by service policy
count	xsd:int	Yes	Maximum number of notifications. For no maximum, either do not specify this part or specify a value of zero.

8.2.1.2   Output message: startNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.2.1.3   Referenced faults

ServiceException from ES 202 504-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0005: Duplicate correlator.

PolicyException from ES 202 504-1 [2]:

- POL0001: Policy error.
- POL0004: Unlimited notifications not supported.
- POL0005: Too many notifications requested.
- POL0009: Invalid frequency requested.

8.2.2   Operation: endNotification

Ends event notifications previously requested by the **startNotification** operation. Content Provider must provide the **correlator** string originally submitted in the **startNotification** operation.

8.2.2.1   Input message: endNotificationRequest

Part Name	Part Type	Optional	Description
correlator	xsd:string	No	Correlator supplied in the **reference** part of the original **startNotificationRequest** message.

8.2.2.2   Output message: endNotificationResponse

Part Name	Part Type	Optional	Description
None			

8.2.2.3 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.

8.3 Interface: StreamingNotification

8.3.1 Operation: notifyMediaStreamRequest

The **notifyMediaStreamRequest** operation is invoked by the Parlay X server when an end-user requests a stream. The notification contains information about the subscriber, the terminal capabilities and the requested stream. To invoke this notification the criteria must match the ones provided in a previous **startNotification** operation.

8.3.1.1 Input message: notifyMediaStreamRequestRequest

Part Name	Part Type	Optional	Description
correlator	xsd:string	No	Identifies the notification request
mediaStreamRequest	MediaStreamRequest	No	A structure containing request details.

8.3.1.2 Output message: notifyMediaStreamRequestResponse

Part Name	Part Type	Optional	Description
None			

8.3.1.3 Referenced faults

None.

8.3.2 Operation: notifyMediaStreamStatus

The **notifyMediaStreamStatus** operation is invoked by the Parlay X server when the status of an ongoing session changes. Session changes are related to the network or terminal status rather than end-user interactions. The information provided could be used by the content provider to change transcoding settings or to cancel an ongoing stream due to changed network capabilities.

8.3.2.1 Input message: notifyMediaStreamStatusRequest

Part Name	Part Type	Optional	Description
correlator	xsd:string	No	Identifies the notification request
sessionIdentifier	xsd:string	No	Session identifier as provided in the initial **notifyMediaStream** or **getStreamingRequests** operations
statusInformation	MediaStreamInformation	No	Communicates status of ongoing media stream and status-specific information.

8.3.2.2 Output message: notifyMediaStreamStatusResponse

Part Name	Part Type	Optional	Description
None			

8.3.2.3 Referenced faults

None.

8.3.3 Operation: notifyError

The error message is sent to the application to indicate that the notification for a session, or for the whole notification, is being cancelled by the Web Service.

8.3.3.1 Input message: notifyErrorRequest

Part name	Part type	Optional	Description
correlator	xsd:string	No	Correlator provided in request to set up this notification.
sessionIdentifier	xsd:string	Yes	The session to which the error applies. If not specified the error applies to all sessions associated with this notification.
reason	common:ServiceError	No	The reason the notification is being discontinued.

8.3.3.2 Output message: notifyErrorResponse

Part name	Part type	Optional	Description
None			

8.3.3.3 Referenced faults

None.

8.3.4 Operation: notifyEnd

The notifications have ended for this **correlator**. This operation will be invoked when the duration or count of notifications has been attained. This operation will not be invoked in the case of an error ending the notifications or deliberate ending of the notification (using **endNotification**).

8.3.4.1 Input message: notifyEndRequest

Part Name	Part Type	Optional	Description
correlator	xsd:string	No	Identifies the notification request

8.3.4.2 Output message: notifyEndResponse

Part Name	Part Type	Optional	Description
None			

8.3.4.3 Referenced faults

None

9   Fault definitions

9.1   ServiceException

9.1.1   SVC0270: Charge failed

This ServiceException is defined in ES 202 504-6 [3].

10   Service policies

Service policies for this service.

Name	Type	Description
Currency	xsd:string	Currency used by service (per ISO 4217 [5])
ContentRequestRetentionTime	common:TimeMetric	A time interval that begins after the receipt of a streaming content request. During this interval, the content request remains available for retrieval by the application.
MaximumNotificationFrequency	common:TimeMetric	Maximum rate of notification delivery (also can be considered minimum time between notifications)
MaximumNotificationDuration	common:TimeMetric	Maximum amount of time a notification may be set up for
MaximumCount	xsd:int	Maximum number of notifications that may be requested
UnlimitedCountAllowed	xsd:boolean	Allowed to specify unlimited notification count (i.e. either by not specifying the optional **count** message part in **startNotificationRequest** or by specifying a value of zero)

Annex A (normative):
WSDL for Multimedia Streaming Control

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_20250419v010101p0.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	