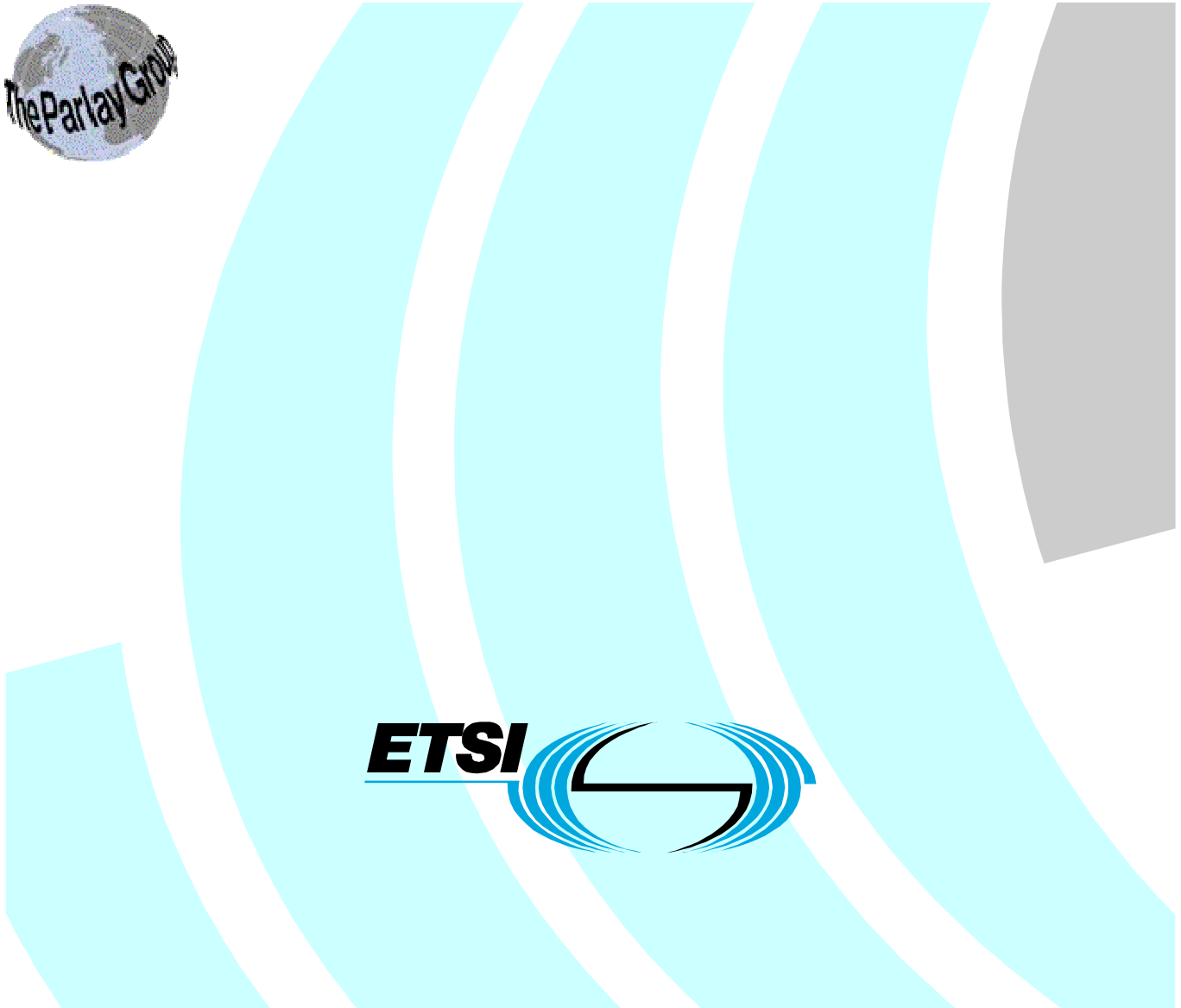
ETSI TR 102 397-2-2 V1.1.1 (2005-08)

Technical Report

Open Service Access (OSA);
Mapping of Parlay X Web Services to Parlay/OSA APIs;
Part 2: Third Party Call Mapping;
Sub-part 2: Mapping to Multi-party Call Control



Reference
DTR/TISPAN-01021-02-02-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

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

The present document is part 2, sub-part 2, of a multi-part deliverable covering Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs, as identified below:

Part 1: "Common Mapping";

Part 2: "Third Party Call Mapping";

Sub-part 1: "Mapping to Generic Call Control";

Sub-part 2: "Mapping to Multi-Party Call Control";

Part 3: "Call Notification Mapping";

Part 4: "Short Messaging Mapping";

Part 5: "Multimedia Messaging Mapping";

Part 6: "Payment Mapping";

Part 7: "Account Management Mapping";

Part 8: "Terminal Status Mapping";

Part 9: "Terminal Location Mapping";

Part 10: "Call Handling Mapping";

Part 11: "Audio Call Mapping";

Part 12: "Multimedia Conference Mapping";

Part 14: "Presence Mapping";

NOTE: Part 13 has not been provided as there is currently no defined mapping between ES 202 391-13 [4] and the Parlay/OSA APIs. If a mapping is developed, it will become part 13 of this series.

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

1 Scope

The Parlay X Web Services provide powerful yet simple, highly abstracted, imaginative, telecommunications functions that application developers and the IT community can both quickly comprehend and use to generate new, innovative applications.

The Open Service Access (OSA) specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the Parlay/OSA APIs.

The present document specifies the mapping of the Parlay X Third Party Call Web Service to the Parlay/OSA Multi-Party Call Control Service Capability Feature (SCF).

2 References

For the purposes of this Technical (TR), the following references apply:

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

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

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

[3] ETSI TR 102 397-1: "Open Service Access (OSA); Mapping of Parlay X Web Services to Parlay/OSA APIs; Part 1: Common Mapping".

[4] ETSI ES 202 391-13: "Open Service Access (OSA); Parlay X Web Services; Part 13: Address List Management".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 102 397-1 [3] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 102 397-1 [3] apply.

4 Mapping description

The Third Party Call capability can be implemented with the Parlay/OSA Multi-Party Call Control SCF. It is applicable to ETSI OSA 1.x/2.x/3.x, Parlay/OSA 3.x/4.x/5.x and 3GPP Releases 4 to 6.

5 Sequence diagrams

5.1 Completed Call Released in Network

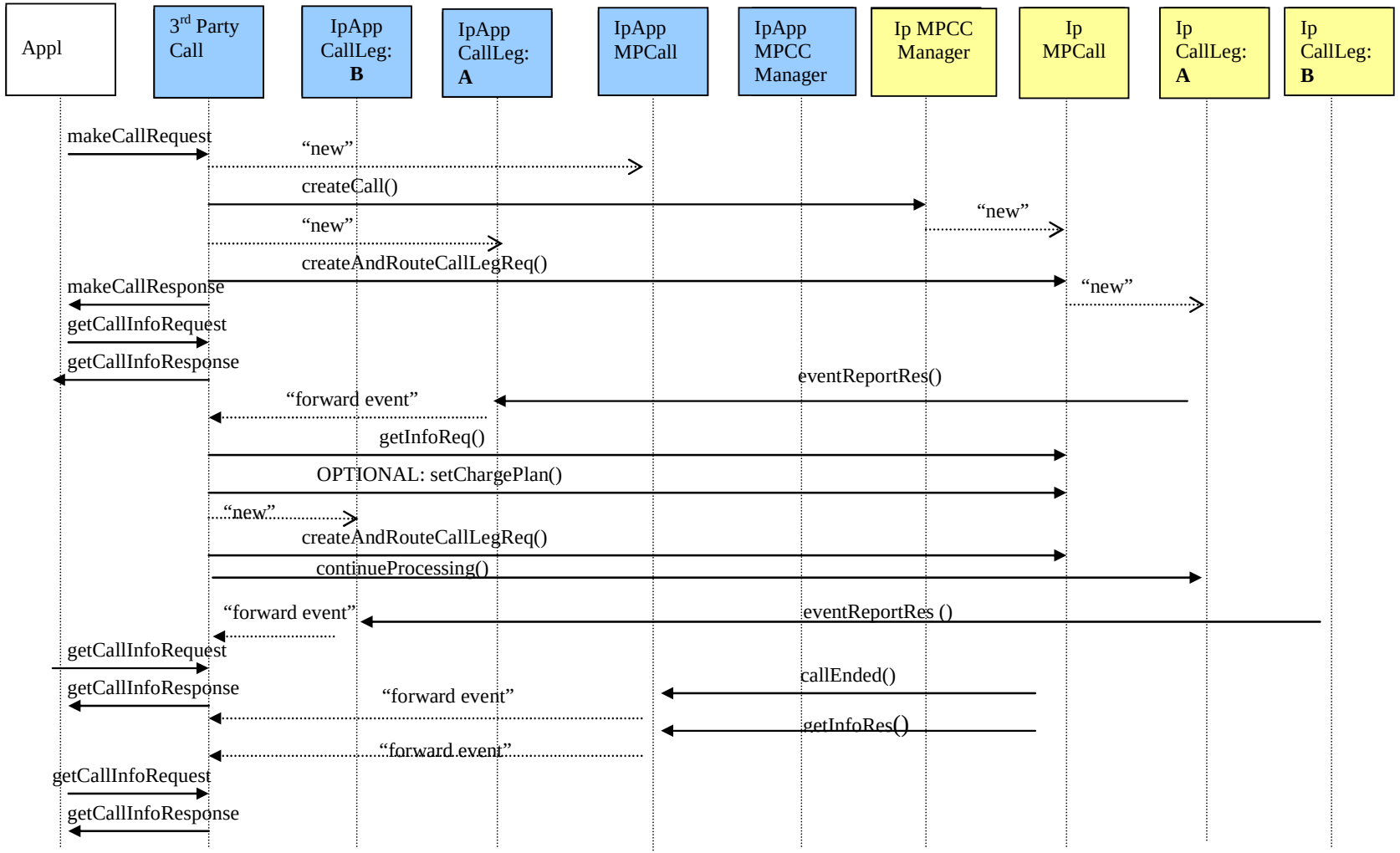


Figure 1

5.2 Completed Call Released by Application

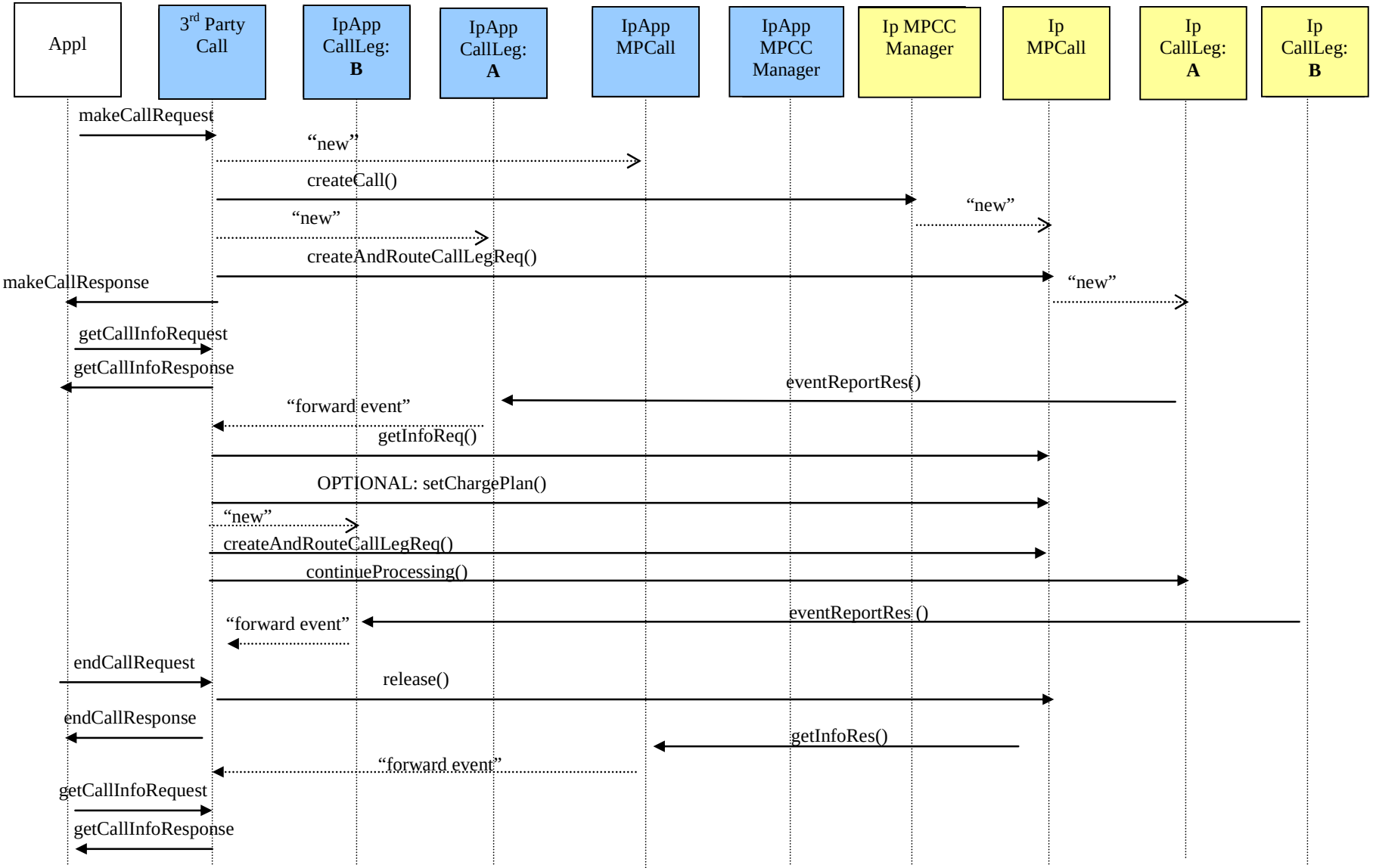


Figure 2

5.3 Call Attempt Abandoned by Application

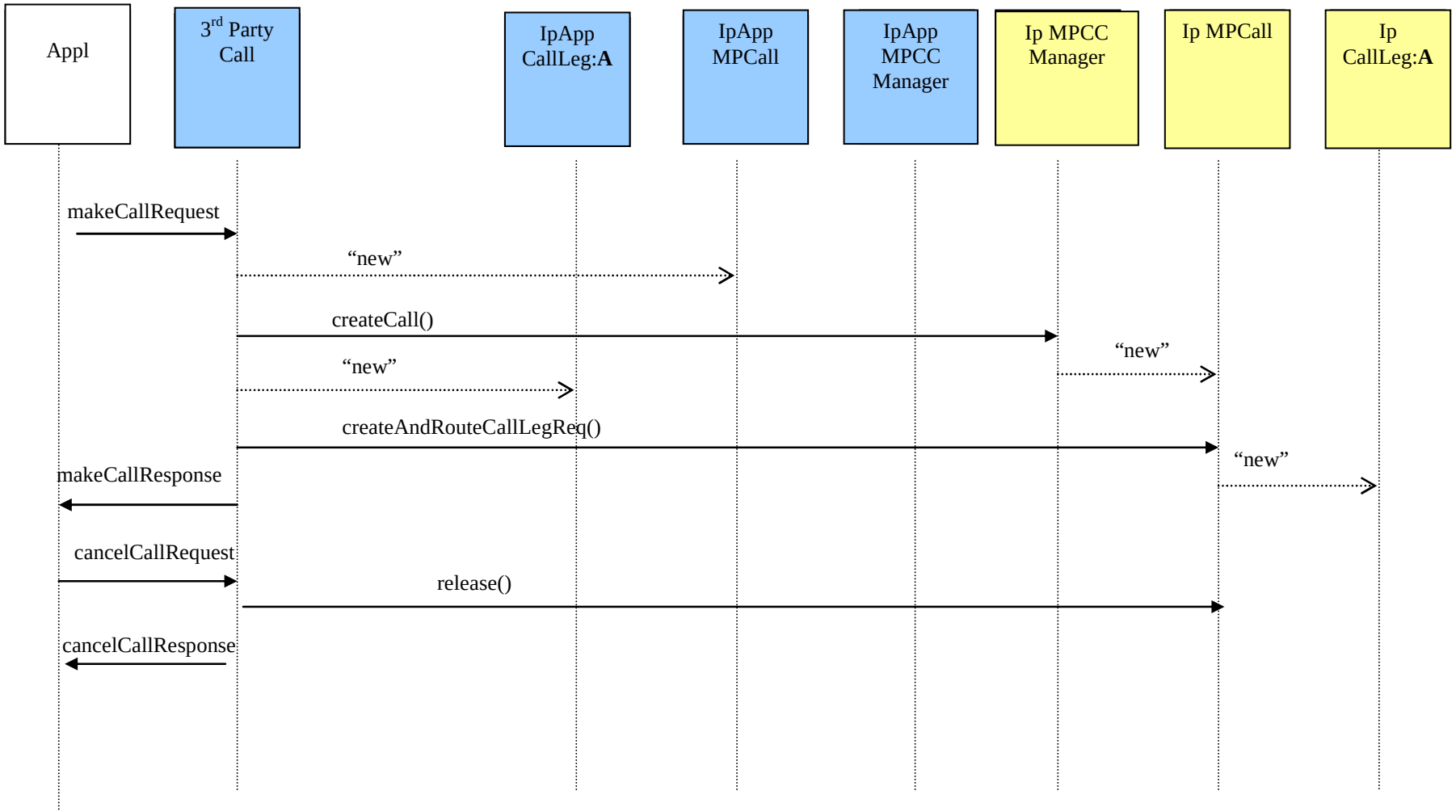


Figure 3

6 Detailed Mapping Information

6.1 Operations

6.1.1 makeCall

The sequence diagram in clause 5.1 illustrates the flow for the **makeCall** operation.

The **makeCall** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- `IpMultiPartyCallControlManager.createCall`
- `IpMultiPartyCall.createAndRouteCallLegReq`
- `IpMultiPartyCall.getInfoReq`
- `IpMultiPartyCall.setChargePlan`

6.1.1.1 Mapping to `IpMultiPartyCallControlManager.createCall`

The `IpMultiPartyCallControlManager.createCall` method is invoked with the following parameters:

Name	Type	Comment
appCall	IpAppMultiPartyCallRef	Reference to callback (internal)

The result from `IpMultiPartyCallControlManager.createCall` is of type `Tp`MultiParty`CallIdentifier` and is used internally to correlate the callbacks. Specifically it is correlated with the value of the **CallIdentifier** part returned to the application in the **makeCallResponse** message

Parlay exceptions thrown by `IpMultiPartyCallControlManager.createCall` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.2 Mapping to IpMultiPartyCall.createAndRouteCallLegReq

The IpMultiPartyCall.createAndRouteCallLegReq method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from IpMultiPartyCallControlManager.createCall]
eventsRequested	TpCallEventRequestSet	Not mapped. [Requests call-related event reports: i.e. including Answer, Busy, No Answer, Not Reachable. The MonitorMode element of each requested event report should have a value of P_CALL_MONITOR_MODE_NOTIFY, with the sole exception of the P_CALL_EVENT_ANSWER event report on the calling party leg, which should have a value of P_CALL_MONITOR_MODE_INTERRUPT]
targetAddress	TpAddress	Specifies the destination leg to which the call should be routed. It is constructed based on the URI provided in the **CallingParty** or **CalledParty** part of **makeCall**, mapped as described in TR 102 397-1 [3]
originatingAddress	TpAddress	Specifies the calling party leg or a third party. To represent the calling party leg, the parameter value is constructed based on the URI provided in the **CallingParty** part of **makeCall**, mapped as described in TR 102 397-1 [3]. To represent a third party (e.g. a service provider number), the parameter may be populated with an address obtained from a pre-defined service level agreement.
appInfo	TpCallAppInfoSet	Not mapped
appLegInterface	IpAppCallLegSet	Not mapped. [Specifies a reference to the application interface that implements the callback interface for the new call leg. Requested events will be reported by the eventReportRes() operation on this interface.]

The result from IpMultiPartyCall.createAndRouteCallLegReq is of type TpCallLegIdentifier and is used internally to correlate the callbacks. It is not mapped to the Parlay X interface.

Parlay exceptions thrown by IpMultiPartyCall.createAndRouteCallLegReq are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.3 Mapping to IpMultiPartyCall.getInfoReq

The IpMultiPartyCall.getInfoReq method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from IpMultiPartyCallControlManager.createCall]
callInfoRequested	TpCallInfoType	Not mapped. [Set to a value of 03h to obtain information on relevant call times and release cause.]

Parlay exceptions thrown by IpMultiPartyCall.getInfoReq are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.1.4 Mapping to IpMultiPartyCall.setChargePlan

The IpMultiPartyCall.setChargePlan method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from IpMultiPartyCallControlManager.createCall]
callChargePlan	TpCallChargePlan	Specifies the charge plan to use. It is constructed based on the values provided in the optional **Charging** part of **makeCall**. See the following table for details.

The `callChargePlan` parameter is constructed as follows:

Name	Type	Comment
ChargeOrderType	TpCallChargeOrderCategory	Not mapped
TransparentCharge	TpOctetSet	Specifies an operator-specific charge plan. It is constructed using the value of the **ChargingInformation.code** element provided in the **Charging** part.
ChargePlan	TpInt32	Not mapped
AdditionalInfo	TpOctetSet	Descriptive string sent to billing system. It is constructed using the value of the **ChargingInformation.description** element provided in the **Charging** part. (May optionally include values of other elements of the **Charging** part.)
PartyToCharge	TpCallPartyToChargeType	Not mapped.
PartyToChargeAdditionalInfo	TpCallPartyToChargeAdditionalInfo	Not mapped

Parlay exceptions thrown by `IpMultiPartyCall.setChargePlan` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.2 getCallInformation

The sequence diagram in clause 5.1 illustrates the flow for the **getCallInformation** operation.

The **getCallInformation** operation is synchronous from the Parlay X client's point of view. It is mapped to the following Parlay/OSA methods:

- `IpAppCallLeg.eventReportRes`
- `IpAppMultiPartyCall.createAndRouteCallLegErr`
- `IpAppMultiPartyCall.getInfoRes`
- `IpAppMultiPartyCall.getInfoErr`
- `IpAppCallLeg.callLegEnded`
- `IpAppMultiPartyCall.callEnded`
- `IpAppMultiPartyCallControlManager.callAborted`

6.1.2.1 Mapping from `IpAppCallLeg.eventReportRes`

The `IpAppCallLeg.eventReportRes` callback method is invoked with the following parameters:

Name	Type	Comment
callLegSessionID	TpSessionID	Not mapped. [The value provide in the result from `IpMultiPartyCall.createandRouteCallLegReq`]
eventInfo	TpCallEventInfo	Specifies the result of the request to route the call to a call party, calling or called, as applicable. See the following discussion for details of its mapping to the **getCallInformation** operation (see note).
NOTE: The `CallMonitorMode` element of `eventInfo` is not mapped.		

If the **getCallInformationRequest** message is received from the application before the `eventReportRes` is invoked for either the Calling Party or Called Party, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallInitial**
- **StartTime**, **Duration**, **TerminationCause** are not applicable

If the **getCallInformationRequest** message is received from the application after the `eventReportRes` is invoked for the Calling Party, and the call attempt was unsuccessful, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of CallTerminated
- **StartTime**, **Duration** are not applicable
- **TerminationCause** has a value mapped from the `CallEventType` element of the `eventInfo` (`TpCallEventInfo`) parameter as follows:
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo element` (of type `TpReleaseCause`) has a value of `P_NO_ANSWER`, then the **TerminationCause** has a value of **CallingPartyNoAnswer**
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo` element (of type `TpReleaseCause`) has a value of `P_BUSY`, then the **TerminationCause** has a value of **CallingPartyBusy**
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo` element (of type `TpReleaseCause`) has a value of `P_NOT_REACHABLE`, then the **TerminationCause** has a value of **CallingPartyNotReachable**
  - if the `CallEventType` has any other value, then the **TerminationCause** has a value of **CallAborted**

If the **getCallInformationRequest** message is received from the application after the `eventReportRes` is invoked for the Called Party, and the call attempt was unsuccessful, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallTerminated**
- **StartTime, Duration** are not applicable
- **TerminationCause** has a value mapped from the `CallEventType` element of the `eventInfo` (`TpCallEventInfo`) parameter as follows:
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo` element (of type `TpReleaseCause`) has a value of `P_NO_ANSWER`, then the **TerminationCause** has a value of **CalledPartyNoAnswer**
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo` element (of type `TpReleaseCause`) has a value of `P_BUSY`, then the **TerminationCause** has a value of **CalledPartyBusy**
  - if the `CallEventType` element has a value of `P_CALL_EVENT_TERMINATING_RELEASE` and the `AdditionalCallEventInfo` element (of type `TpReleaseCause`) has a value of `P_NOT_REACHABLE`, then the **TerminationCause** has a value of **CalledPartyNotReachable**
  - if the `CallReportType` has any other value, then the **TerminationCause** has a value of **CallAborted**

If the **getCallInformationRequest** message is received from the application while the call between the two Parties is Active (i.e. after the `eventReportRes` is invoked for the Called Party, where the `CallEventType` element of the `eventInfo` parameter has a value of `P_CALL_EVENT_ANSWER`), then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallConnected**
- **StartTime** has the value provided in the `CallEventTime` element of the `eventInfo` parameter
- **Duration** and **TerminationCause** are not applicable

6.1.2.2 Mapping from IpAppMultiPartyCall.createAndRouteCallLegErr

If the **getCallInformationRequest** message is received from the application before the createAndRouteCallLegErr is invoked for either the Calling Party or Called Party, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallInitial**
- **StartTime**, **Duration**, **TerminationCause** are not applicable

If the **getCallInformationRequest** message is received from the application after the createAndRouteCallLegErr is invoked for either the Calling or Called Party, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallTerminated**
- **StartTime**, **Duration** are not applicable
- **TerminationCause** has a value of **CallAborted**.

6.1.2.3 Mapping from IpAppMultiPartyCall.getInfoRes

The IpAppMultiPartyCall.getInfoRes callback method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from IpMultiPartyCallControlManager.createCall]
callInfoReport	TpCallInfoReport	Specifies the call information requested prior to call completion. See the following discussion for details of its mapping to the **getCallInformation** operation (see note).
NOTE: The CallInfoType, CallInitiationStartTime, CallConnectedToResourceTime elements of callInfoReport are not mapped.		

If the **getCallInformationRequest** message is received from the application before the IpAppMultiPartyCall.getInfoRes method is invoked, then the **CallInformation** part of the **getCallInformationResponse** message is constructed using information received from previously invoked callback methods: e.g. IpAppCallLeg.eventReportRes.

If the **getCallInformationRequest** message is received from the application after invocation of the IpAppMultiPartyCall.getInfoRes method and within a time interval following the termination of the call between the two Parties, where this time interval is defined by the **StatusRetentionTime** service policy, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallTerminated**
- **StartTime** has the value provided in the CallConnectedToDestinationTime element of the callInfoReport parameter.
- **Duration** has a value derived by subtracting the values of the CallEndTime and CallConnectedToDestinationTime elements of the callInfoReport parameter.
- **TerminationCause** has a value derived from the Cause element of the callInfoReport parameter. If the Cause element has a value that explicitly indicates that one of the call parties disconnected to terminate the call, then **TerminationCause** has a value of **CallHangUp;** otherwise, **TerminationCause** has a value of **CallAborted.**

6.1.2.4 Mapping from IpAppMultiPartyCall.getInfoErr

If the **getCallInformationRequest** message is received from the application either before or after the getInfoErr method is invoked, then the **CallInformation** part of the **getCallInformationResponse** message is constructed using information received from previously invoked callback methods, if any: e.g. IpAppCallLeg.eventReportRes.

6.1.2.5 Mapping from IpAppCallLeg.callLegEnded

If the **getCallInformationRequest** message is received from the application after the `callLegEnded` method is invoked, then the **CallInformation** part of the **getCallInformationResponse** message is constructed using information received from previously invoked callback methods: e.g. `IpAppMultiPartyCall.getInfoRes/Err`.

6.1.2.6 Mapping from IpAppMultiPartyCall.callEnded

The `IpAppMultiPartyCall.callEnded` callback method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from `IpMultiPartyCallControlManager.createCall`]
report	TpCallEndedReport	Specifies the reason the call is terminated. See the following discussion for details of its mapping to the `getCallInformation` operation (see note).
NOTE: The `CallLegSessionID` element of report is not mapped.		

If the **getCallInformationRequest** message is received from the application within a time interval following the termination of the call between the two Parties, where this time interval is defined by the **StatusRetentionTime** service policy, then the **CallInformation** part of the **getCallInformationResponse** message is constructed as follows:

- **CallStatus** has a value of **CallTerminated**
- **StartTime.** If the `IpAppMultiPartyCall.getInfoRes` method has already been invoked, then the value is derived from the `CallConnectedToDestinationTime` element of the `callInfoReport` parameter of the `IpAppMultiPartyCall.getCallInfoRes method`. Otherwise the value is the date/time that the `eventReportRes` was received for the Called Party: i.e. where the `CallEventType` element of the eventInfo parameter has a value of `P_CALL_EVENT_ANSWER`.
- **Duration.** If the `IpAppMultiPartyCall.getInfoRes` method has already been invoked, then the value is derived by subtracting the values of the `CallEndTime` and `CallConnectedToDestinationTime` elements of the `callInfoReport` parameter of the `IpAppMultiPartyCall.getInfoRes` method. Otherwise, the value is derived by subtracting the value of **StartTime** from the date/time that the notification of call termination was received from the network, i.e. invocation of `IpAppMultiPartyCall.callEnded`.
- **TerminationCause** has a value derived from the `Cause` element of the report parameter of the `IpAppMultiPartyCall.callEnded method`. [Alternatively, if the `IpAppMultiPartyCall.getInfoRes` method has already been invoked, this information may be derived from the `Cause` element of the `callInfoReport` parameter of the `IpAppMultiPartyCall.getInfoRes method`.] In either case, if the `Cause` element is provided and it has a value that explicitly indicates that one of the call parties disconnected to terminate the call, then **TerminationCause** has a value of **CallHangUp;** in all other cases, **TerminationCause** has a value of **CallAborted.**

6.1.2.7 Mapping from IpAppMultiPartyCallControlManager.callAborted

If the **getCallInformationRequest** message is received from the application after the `callAborted` method is invoked, then the **CallInformation** part of the **getCallInformationResponse** message is constructed using information received from previously invoked callback methods: e.g. `IpAppCallLeg.eventReportRes`.

6.1.3 endCall

The sequence diagram in clause 5.2 illustrates the flow for the **endCall** operation.

The **endCall** operation is synchronous from the Parlay X client's point of view. It is mapped to the Parlay/OSA method: `IpMultiPartyCall.release`. If the **endCallRequest** message is received from the application after the referenced call has been terminated, it is not mapped to `IpMultiPartyCall.release`; instead a Parlay X exception is thrown: **SVC0261**.

The `IpMultiPartyCall.release` method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from `IpMultiPartyCallControlManager.createCall`]
cause	TpReleaseCause	Assigned a value indicating application-initiated call termination

Parlay exceptions thrown by `IpMultiPartyCall.release` are mapped to Parlay X exceptions as defined in clause 6.2.

6.1.4 cancelCall

The sequence diagram in clause 5.3 illustrates the flow for the **cancelCall** operation.

The **cancelCall** operation is synchronous from the Parlay X client's point of view. It is mapped to the Parlay/OSA method: `IpMultiPartyCall.release`. If the **endCallRequest** message is received from the application after the referenced call has been connected, it is not mapped to `IpMultiPartyCall.release`; instead a Parlay X exception is thrown: **SVC0260**.

The `IpMultiPartyCall.release` method is invoked with the following parameters:

Name	Type	Comment
callSessionID	TpSessionID	Not mapped. [The value provide in the result from `IpMultiPartyCallControlManager.createCall`]
cause	TpCallReleaseCause	Assigned a value indicating application-initiated call termination

Parlay exceptions thrown by `IpMultiPartyCall.release` are mapped to Parlay X exceptions as defined in clause 6.2.

6.2 Exceptions

For the present document, the mapping of Parlay/OSA API method exceptions to Parlay X Web Service exceptions is common and defined in TR 102 397-1 [3]. There are no service-specific exception mappings.

7 Additional Notes

No additional notes are provided.

History

Document history		
V1.1.1	August 2005	Publication