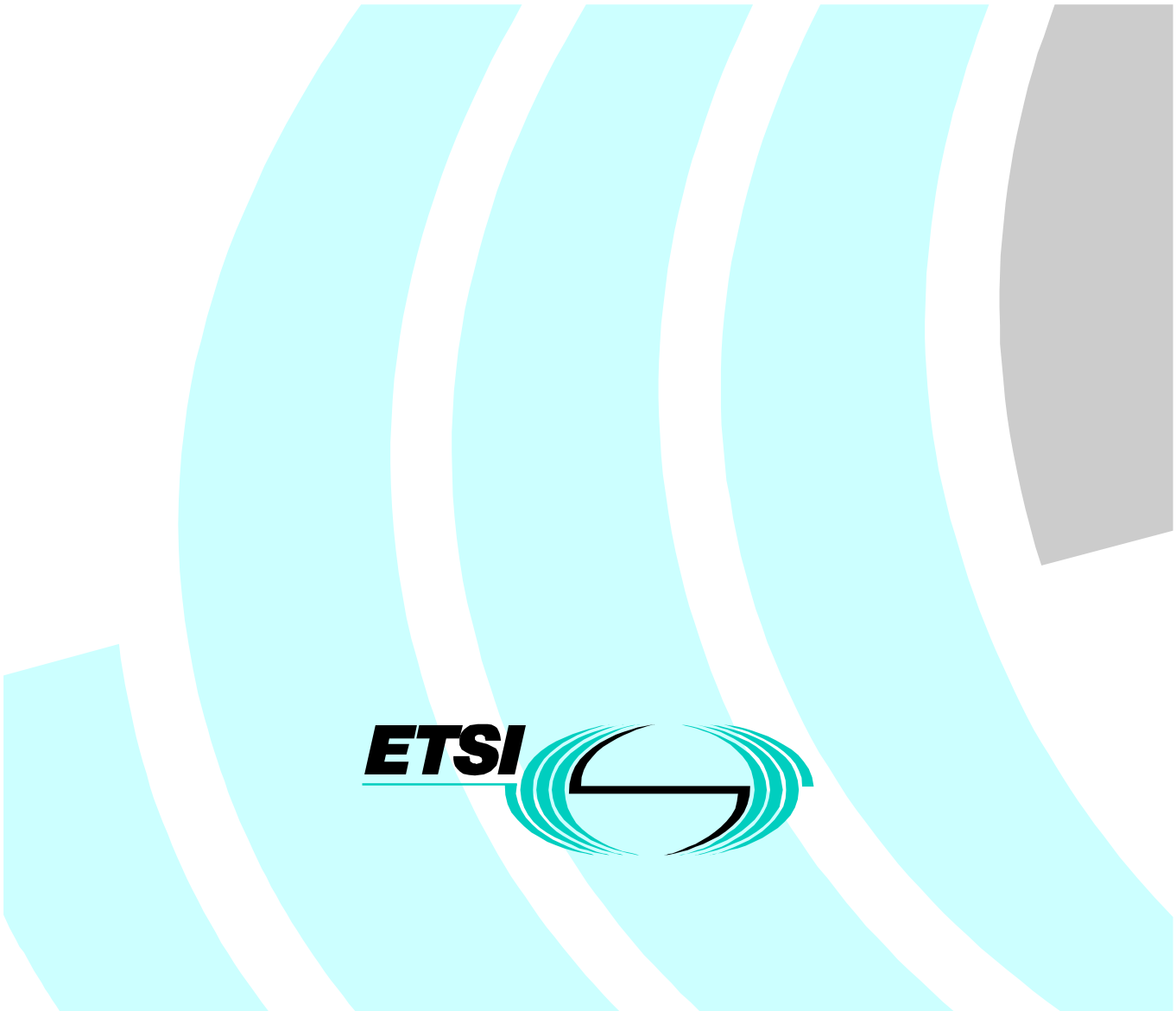
ETSI EG 201 899 V1.1.1 (2001-05)

ETSI Guide

Services and Protocols for Advanced Networks (SPAN);
Service Provider Access;
Modelling service provider access requirements using an API approach



Reference
DEG/SPAN-140504

Keywords
API, architecture, interface, UML

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://www.etsi.org/ipr).

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 Guide (EG) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN).

1 Scope

The present document applies to the first phase of the Service Provider Access Requirements (SPAR) work, aiming primarily at fixed PTNs, e.g. public switched telecommunications networks (PSTNs) and Integrated Services Digital Networks (ISDNs). This first phase is described by two documents, Service Provider Access Requirements; Enhanced Telephony Services [1] and Network operators requirements for the delivery of service providers access [2]. The present document shows how these requirements can be fulfilled using an Application Programming Interface (API).

2 References

The following documents contain provisions, which through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] ETSI EG 201 722: "Intelligent Network (IN); Service provider access requirements; Enhanced telephony services".

[2] ETSI EG 201 807: "Network Aspects (NA); Intelligent Network (IN); Network operators' requirements for the delivery of service provider access".

[3] ISO/IEC JTC 1 Directives. Supplement 2: Guidelines for API Standardization.

[4] ETSI ETS 300 335: "Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 1; Test specification".

[5] ETSI ETS 300 090 (1992): "Integrated Services Digital Network (ISDN); Calling Line Identification Restriction (CLIR) supplementary service; Service description".

[6] ETSI ETR 322: "Intelligent network (IN); Vocabulary of terms and abbreviations for CS-1 and CS-2".

[7] ETSI TS 123 127: "Universal Mobile Telecommunications System (UMTS); Virtual Home Environment/Open Service Architecture (3GPP TS 23.127 version 3.2.0 Release 1999)".

[8] ITU-T Supplement 29 to Series Q. Recommendations: "Service Modelling Evolution to the Use of Object Oriented Techniques".

[9] ITU-T Recommendation Q.65 (2000): "The unified functional methodology for the characterization of services and network capabilities including alternative object-oriented techniques".

[10] ETSI ETS 300 128 (1992): "Integrated Services Digital Network (ISDN); Malicious Call Identification (MCID) supplementary service; Service description".

3 Definitions and abbreviations

3.1 Definitions

For the purpose of the present document, the following terms and definitions apply:

application: entity in a service provider's domain that provides a service (see definition of **service** below)

application programming interface: boundary across which application software uses facilities of programming languages to invoke services (see ISO/IEC JTC 1 Directives)

calling line identity: number that uniquely identifies a subscriber line that is used for a call <Add Citation [13]>

end user: see "**service user**" definition

network API gateway: API entity that provides access to the public telecommunications network

network-network interface: interface at a network node which is used to interconnect the node with another network node (see EG 201 722)

network-provided calling line identity: that is provided by the originating public telecommunications network to a call set-up request, if the calling party has not provided any calling line identity or the user-provided calling line identity has not passed a verification in the network (see ETS 300 335)

presentation-restricted calling line identity: calling line identity that is associated with a marking informing the terminating local exchange not to display this calling line identity to the called party (see ETS 300 090)

public telecommunications network: telecommunications network which provides telecommunications services to the general public (see ETR 322)

public telecommunications network operator: entity which is responsible for the development, provisioning and maintenance of telecommunications services to the general public and for operating the corresponding networks (see ETR 322)

service: that which is offered by an administration or recognized private operating agency (i.e. a public or private service provider) to its customers in order to satisfy a telecommunication requirement (see ETR 322)

service capability feature: functionality offered by service capabilities that are accessible via the standardized OSA interface (see TS 123 127)

service provider: entity which provides services to its service subscribers on a contractual basis and who is responsible for the services offered. The same organization may act as a public telecommunications network operator and a service provider (see ETR 322)

service provider access: access facility that enables a service provider to access specific functionality of a public telecommunications network (see EG 201 722)

service provider access interface: interface between a public telecommunications network and a service provider's equipment for enabling the service provider to access specific functionality of a public telecommunications network (see EG 201 722)

service provider access requirement: requirement for access by a service provider to specific functionality of a public telecommunications network (see EG 201 722)

service subscriber: entity that contracts for services offered by service providers (see ETR 322)

service user: entity external to the network that uses the services offered by the PTNO or SP (see ETR 322)

user-network interface: interface between the terminal equipment and a network termination point at which the access protocols apply (see EG 201 722)

user-provided calling line identity: network number that has been provided by the calling party (see ETS 300 335)

user-provided, not screened calling line identity: network number that has been provided by the calling party and has been passed forward by the originating public telecommunications network without performing any screening function for verification purposes (see ETS 300 335)

user-provided, verified and passed calling line identity: network number that has been provided by the calling party and has been successfully verified in the originating public telecommunications network (see ETS 300 335)

3.2 Abbreviations

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

3GPP	Third Generation Partnership Project
API	Application Programming Interface
CAMEL	Customized Applications for Mobile Enhanced Logic
CAP	CAMEL Application Part
CLI	Calling Line Identity
CS-1	Capability Set One
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU-T	International Telecommunications Union - Telecommunication standardization sector
IVR	Interactive Voice Response
MCID	Malicious Call Identification
NNI	Network-Network Interface
NRA	National Regulatory Authority
NTP	Network Termination Point
O-O	Object Oriented
OSA	Open Service Architecture
PIN	Personal Identification Number
PoP	Point of Presence
PSTN	Public Switched Telephone Network
PTN	Public Telecommunications Network
PTNO	Public Telecommunications Network Operator
SCF	Service Control Function
SMTP	Simple Mail Transmission Protocol
SP	Service Provider
SPA	Service Provider Access
SPAI	Service Provider Access Interface
UI	User Interaction
URL	Universal Resource Locator
UML	Unified Modelling Language
UNI	User-Network Interface

4 Introduction

An overview of an API is presented in clause 5, and mappings between the service provider access requirements and the API methods are presented in clause 6. Sequence diagrams for example services are available in APIs for third party service applications ES 201 915, and further information on service modelling is available in ITU-T documentation [8] and [9].

API ES 201 915 implementation issues, which are not covered in the present document, include the following:

- dimensioning and scalability of operations;
- number and efficiency of operations;
- object management;
- error control, including error detection and recovery, needs to be considered for all operations over the API. Those error control requirements specifically defined in the Service provider access requirements; Enhanced telephony services [1] have been addressed in the API mapping examples in the present document, but other error control mechanisms are defined in APIs for third party service applications ES 201 915;
- some of the requirements defined in the Service provider access requirements; Enhanced telephony services [1] may require access to PTN hosted user profile information, e.g. incoming call barring and call diversion information. The means of accessing such information is not defined in the present document. This is a management plane requirement, as shown in [1];
- the call leg methods are not supported in the 3GPP OSA Release '99 [7].

5 Architecture

5.1 API overview

An Application Programming Interface (API) is a means of supporting the service provider access requirements identified by ETSI. This API is defined using the Unified Modelling Language (UML), which is an Object Oriented (O-O) technique [8] and [9].

Detailed descriptions of the API can be found in ETSI documentation ES 201 915, but an overview of the API architecture is shown in figure 1.

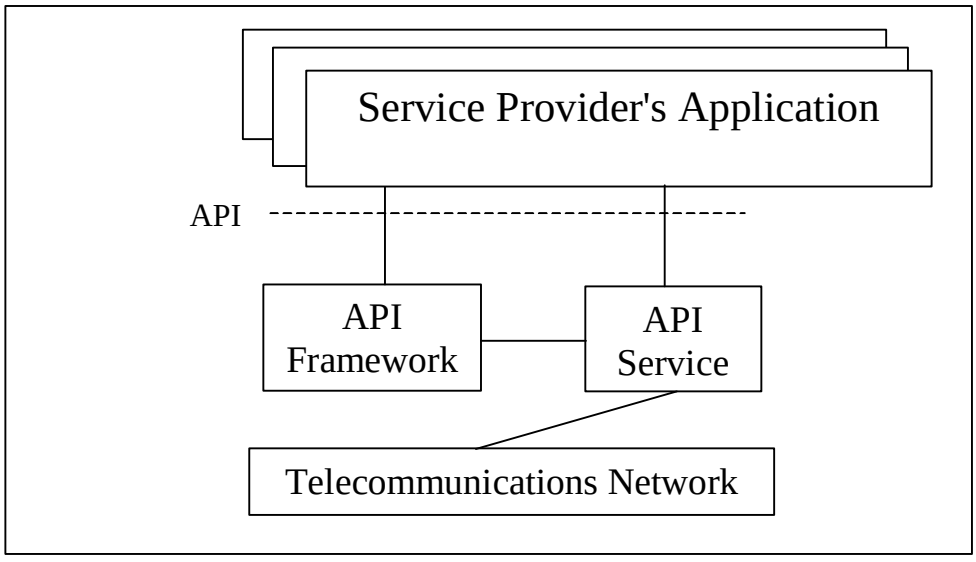


Figure 1: API Architecture

Service providers' applications (telecommunications services) and the PTN's service capability features use the API to utilize each others resources in delivering services to end users. A contractual agreement and physical access interface must exist between the service provider and network operator for the API to be used, the nature of these is dependent upon commercial agreements between the parties involved and may be subject to national regulations.

This API, which is based on the ETSI API ES 201 915, contains an API Framework and one or more API Service, where the API Framework provides control of the service-surround capabilities, and API Service provide control of the real time network capabilities.

API Framework includes the following:

- Trust and security management, which is used to make initial contact with a framework provider, to authenticate the application and framework, and to obtain access to other framework interfaces and service capability features. This includes service capability feature selection and electronically signing service agreements;
- Service Discovery, which is used by the Application to get the identities of the service capability features that are available from the underlying network. The identities are used during authentication to select the required service capability features;
- Integrity Management, which is used for load management, fault management, heartbeat management, heartbeat and Operation, Administration and Maintenance;
- Service Subscription, which is used to manage the contractual agreement between the parties.

API Service include:

- Generic Call Control Service, which is used to enable and disable call notifications, to notify call events, and to control and manage simple calls consisting of one or two legs;
- INAP-1 Call Control Service (INAP CS-1), which enhances the Generic Call Control Service by allowing more complex call behaviour to be used;
- CAP Call Control Service, which enhances the Generic Call Control Service to allow more control over charging for mobile terminals;
- Enhanced Call Control Service, which enhances the Generic Call Control Service with load control, overload reporting, and call leg control;
- Multi Media Call Control Service, which enhances the Enhanced Call Control Service to allow the control of specific media channel characteristics;
- Conference Call Control Service, which enhances the Enhanced Call Control Service to allow the creation of sub-conferences and the ability to move legs between sub-conferences, or merge sub-conferences;
- Multi Media Conference Call Service, which enhances the Conference Call Control and Multi Media Call Control Services to allow interworking with network signalled conference protocols, manipulation of media, and the handling of multi media conference policies;
- Generic Messaging Service, which is used to send, store and receive electronic and voice mail;
- User Location Service, which is used to establish the geographic location and status of fixed, mobile and IP-based telephony users;
- User Location CAMEL Service, which supplements the User Location Service to provide network related information;
- User Location Emergency Service, which supplements the User Location Service with specialized functionality to handle emergency calls;
- User Status Service, which is used to request the current status, or the reporting of a change of status, of fixed, mobile or IP-based telephony users;

- Generic User Interaction Service, which is used to interact with end users;
- Call User Interaction Service, which is used in conjunction with another service interface (only the Generic Call Control Service at present) to send information to, or gather information from, an end user.

Detailed example sequence diagrams can be found in APIs for third party service applications ES 201 915, but an overview is given below.

When an Application starts, the following is a typical outline sequence of events:

1) application initiates client authentication;
2) mutual authentication of client and Framework Interface;
3) application discovers service capability features;
4) application selects required service capability features;
5) mutual signing of service agreement, e.g. by electronic signature.

At this point the agreed network operator service capability features become available to the Application. If the agreed service capability features include inbound calls, the Application requests call notifications on the appropriate Service Interface. The service capability features, and any call notifications, remain active until one of the parties chooses to either terminate the service agreement, or disable the call notifications.

If call notifications have been enabled, the following is a typical outline sequence of events:

1) a call arrives in network and is identified as requiring a service provider;
2) a call object is created. (Other objects may also be created, e.g. Call leg object);
3) A call event notification is sent to the appropriate Application over the API;
4) the Application sends all appropriate information for call treatment, including routing and reporting requirements;
5) network applies the call treatment and reports outcome, if requested;
6) the Application provides subsequent instructions, if required;
7) the network provides subsequent reports, if and when required.

The above sequence is repeated for each call received and an unlimited number of calls may co-exist.

If the Application initiates calls, the following is a typical outline of events:

1) the Application creates a call object. (Other objects may also be created);
2) the Application sends all appropriate information for call routing and reporting;
3) the network routes the call and provides requested reports;
4) the Application provides subsequent instructions, if required;
5) the network provides subsequent reports, if required.

The above sequence is repeated for each call initiated and an unlimited number of calls may co-exist.

5.2 Description of classes

Framework and service interface classes are defined in the ETSI document: APIs for third party service applications ES 201 915.

5.3 Interface class diagrams

Interface class diagrams are defined in the ETSI document: APIs for third party service applications ES 201 915.

6 An API approach to modelling the SPA requirements

6.1 Introduction

Information flows and API mapping examples for each of the service provider phase 1 requirements, as defined in Service Provider Access Requirements; Enhanced Telephony Services [1], are presented below. Only the mappings, parameters and their values that directly relate to the individual requirements are shown in this clause. The complete definitions of the API methods, parameters and parameter values can be found in the API Specifications ES 201 915.

The API mapping examples may be modified by the appropriate standardization groups and the regulatory considerations have been included for information only. The network operators' requirements, as defined in Network operators' requirements for the delivery of service provider access [2], have been included where appropriate. The screening and charging aspects of the network operators document ([2] clauses 5.5.1 and 5.5.2, respectively) apply to all the requirements and have not been shown, as these have no direct impact on the information flows between service providers and network operators.

The circuit related and non-circuit related requirements have been combined to show the logical information flows, which make no assumptions about the actual implementation. In all cases below, the application is assumed to have been authenticated.

It should be noted that the API methods fall into two categories, asynchronous and synchronous. Asynchronous methods, which are identified by the suffix 'Req' (Request), may receive explicit responses identified with the suffixes 'Err' (Error) or 'Res' (Result). Synchronous methods, which have no suffix, may receive implicit responses (return values) when the methods terminate. These implicit responses are not shown in the API mapping examples.

PTNs will need the ability perform functions not defined in the network operators requirements [2], e.g. to monitor calls for charging, operational and regulatory reasons, and to perform legal intercept, which are not included in the mapping examples.

6.2 Calling party information handling capabilities

The implementation of the service provider functional requirements relating to the CLI should be in conformance with the general European Commission and national regulations and with bilateral agreements where they exist.

6.2.1 Reception of the calling line identity

The service provider needs to receive the Calling Line Identity (CLI) from the PTN with a call or call indication, if the CLI is available in the PTN in accordance with [1] clause 5.2.1 and [2] clauses 5.2.1 and 5.2.2. If the calling party is using the services of the service provider and national regulations and/or legislation allows it, this requirement also applies to a CLI marked as 'presentation-restricted'. All the indicators associated with the CLI need also to be delivered.

Information flows:

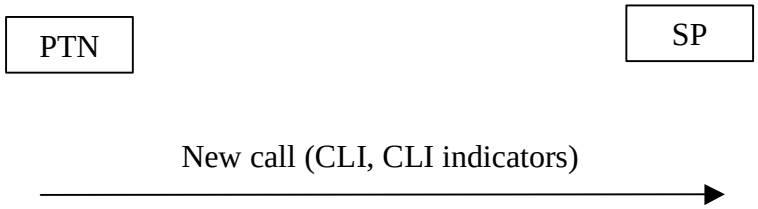


Figure 2: Reception of the calling line identity

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification () and CLIs are supported in the underlying PTN.

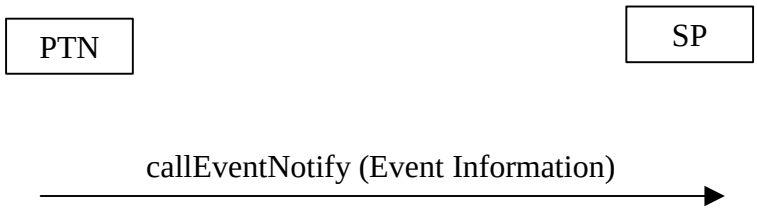


Figure 3: Reception of the calling line identity

The Event Information includes the parameters shown in table 1.

Table 1: Event Information parameters

Destination Address	See table 2.
Originating Address	See table 2.
Original Destination Address	See table 2.
Redirecting Address	See table 2.
Call Application Information	Alerting mechanism, Network access type, Interworking indicators, Teleservice, Bearer service, Party category, Presentation address (see table 2), Generic information, Additional address.
Call Event Name	Off hook, Address information collected, Address information is analysed, Called party is busy, Called party is unreachable, No answer from called party, Failure in routing the call.

Regulatory considerations:

National regulatory authorities may impose restrictions on the delivery of some CLI information, therefore the PTNO requires the ability to prevent the forwarding of restricted information.

6.2.2 Presentation of the complete CLI information to the PTN

The service provider needs the ability to present all the CLI information about the calling party to the PTN in accordance with [1] clause 5.2.2. This includes the original CLI together with the related status information, as well as all the relevant indicators of the category of the call.

Information flows:

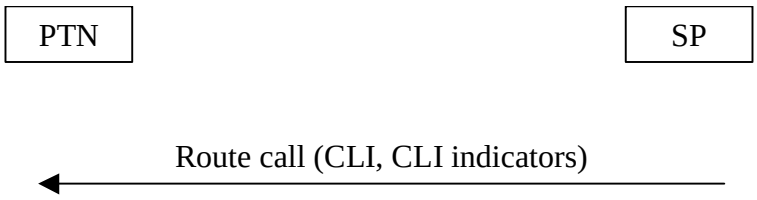


Figure 4: Presentation of the complete CLI information to the PTN

API mapping example:

Pre-conditions: A call object exists and the service provider has either received a callEventNotify (Event Information), or is to initiate a call. CLIs are supported in the underlying PTN. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

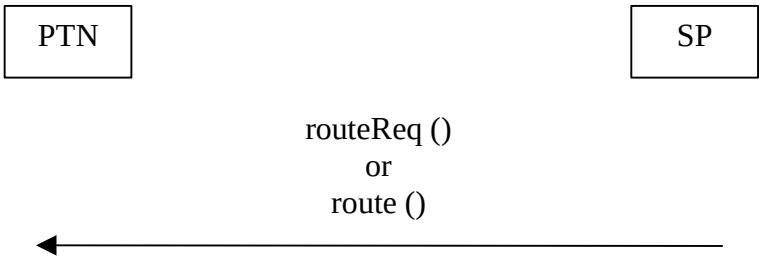


Figure 5: Presentation of the complete CLI information to the PTN

The routeReq () method includes address parameters, which include the elements shown in table 2.

Table 2: Address parameter elements.

Address Plan	IP, E.164, E.164 Mobile, URL, SMTP, Undefined.
Address String	Address required by the service provider, e.g. CLI, dialled number, destination address.
Name	Name associated with Address String.
Address Presentation	'Presentation allowed', 'presentation restricted', or 'presentation address not available' (see note).
Address Screening	'User provided verified and passed', 'user provided not verified', 'user provided verified and failed', or 'application/network provided'.
Subaddress String	Subaddress.
NOTE: The address parameters may be null if the address is not available.	

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission).

Regulatory considerations:

National regulatory authorities may impose restrictions on the use of CLI information supplied by service providers. It is assumed that the CLI provided by the PTN will not be changed, although the CLI provided by the service provider may be used for presentation to the called party. Services such as Malicious Call Identification (MCID) [10] must not be compromised.

6.2.3 Addition or substitution of a calling line identity

The service provider needs the ability to add or substitute the User Provided CLI to the CLI information of a call when passing it forward in accordance with [1] clause 5.2.3.

Information flows:

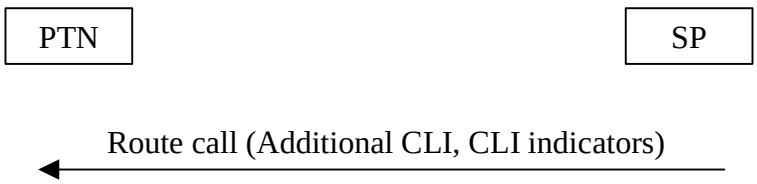


Figure 6: Addition or substitution of a calling line identity

API mapping example:

Pre-conditions: A call object exists and the service provider has either received a callEventNotify (Event Information), or is to initiate a call. A second CLI can be transported through the underlying PTN and delivered to the service user. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

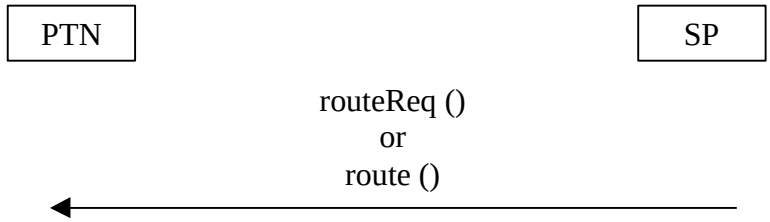


Figure 7: Addition or substitution of a calling line identity

The routeReq () method includes address parameters, which include the elements shown in table 2.

Regulatory considerations:

National regulatory authorities may impose restrictions on the use of CLI information supplied by service providers. In general, it is assumed that the CLI provided by the PTN will not be changed, although the CLI provided by the service provider may be used for presentation to the called party. Services such as Malicious Call Identification (MCID) [10] must not be compromised.

6.2.4 Provision of CLI information to an SP-initiated call

The service provider needs the ability to provide CLI information to an SP-initiated call in accordance with [1] clause 5.2.4.

Information flows:

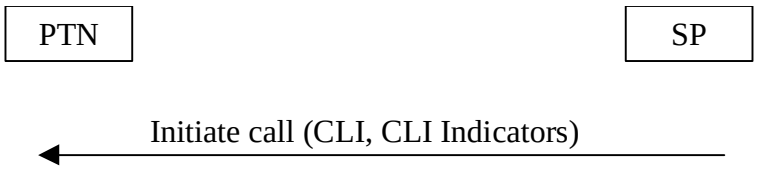


Figure 8: Provision of CLI information to an SP-initiated call

API mapping example:

Pre-conditions: A call object has been created and CLIs are supported in the underlying network. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

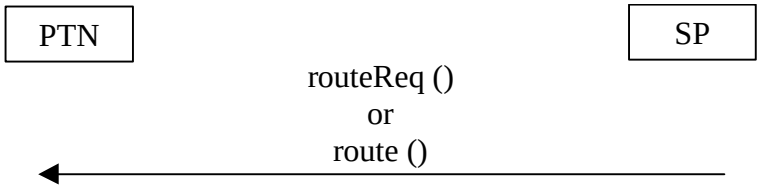


Figure 9: Provision of CLI information to an SP-initiated call

The routeReq () method includes address parameters, which include the elements shown in table 2.

Regulatory considerations:

National regulatory authorities may impose restrictions on the use of CLI information supplied by service providers.

6.2.5 Relaying of the malicious call identification data of a received call

The service provider needs the ability to relay all the received call-specific information unchanged to the terminating network, that may be used for malicious call tracing in accordance with [1] clause 5.2.5 and [2] clause 5.2.1.

Information flows:

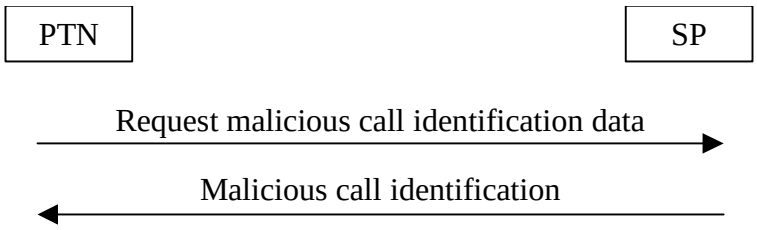


Figure 10: Relaying of the malicious call identification data of a received call

NOTE: The request information flow is only required if the data has not been supplied during the call set-up phase.

API mapping example:

The malicious call identification data can be supplied during the call set-up phase, as shown below, but there is no support for requesting this information.

Pre-conditions: A call object exists and the service provider has either received a callEventNotify (Event Information), or is to initiate a call. The PTNO has mechanisms in place to prevent essential MCID information being overwritten in the PTN by the SP. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

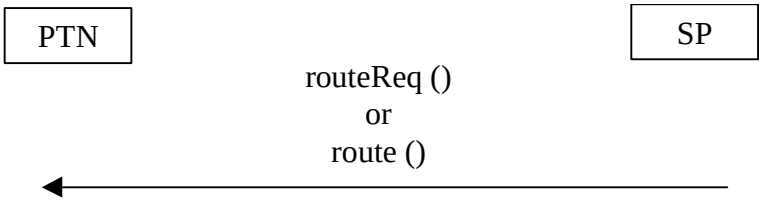


Figure 11: Relaying of the malicious call identification data of a received call

The routeReq() and createCallLeg () methods include Origination Address, Redirecting Address and Call Application Information, including a Presentation Address, parameters, which should contain sufficient addressing information for malicious call identification when combined with information available in the network API gateway. The address parameters include the elements shown in table 2. The Call Application Information is shown in table 1.

Regulatory considerations:

National regulatory authorities are likely to require the PTN to provide malicious call identification information for a single call to support the malicious call identification service (MCID) [10]. In cases where a call has been routed through a service provider's equipment as two separate calls, e.g. incoming and outgoing calls joined together by the service provider, the malicious call identification information held by the PTN will not associate the two calls together.

6.3 Basic call set-up and clear-down capabilities

6.3.1 Return speech path connection from the terminating PTN to the calling party

The service provider needs the ability to request the through-connection of a backward in-band message path to the calling party immediately upon the arrival of a confirmation of the call set-up in accordance with [1] clause. 5.3.1. There must be a mechanism for the support of charging between the PTNO and the service provider.

Information flows:

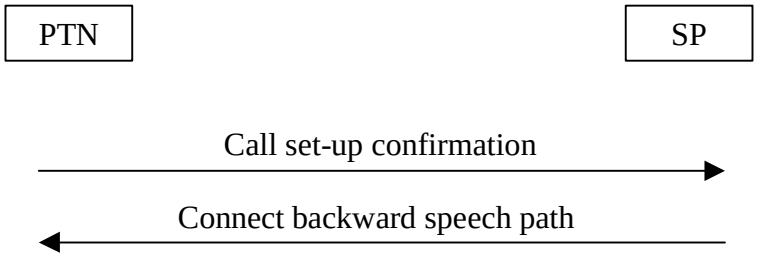


Figure 12: Return speech path connection from the terminating PTN to the calling party

API mapping example:

Pre-conditions: A call set-up has been attempted using routeReq (responseRequested) or createCallLeg (), eventReportReq (eventReportsRequested) and route (). The resources to provide the backward speech path must be available in the underlying PTN.

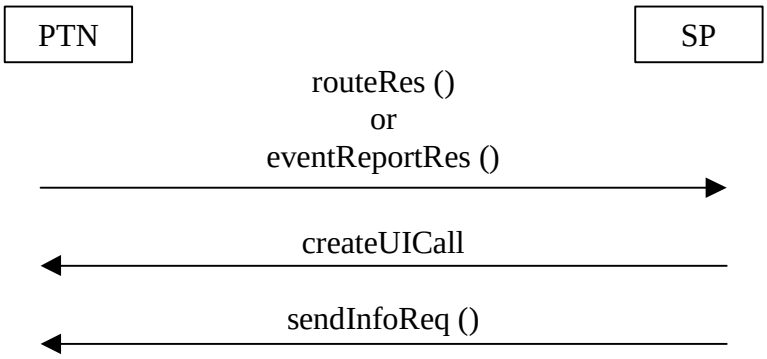


Figure 13: Return speech path connection from the terminating PTN to the calling party

The sendInfoReq () includes the parameters shown in table 3.

Table 3: sendInfoReq () parameters

Information	Specifies the identity of the information to send to the user.
Variable Information	Defines the variable part of the information to send to the user.
Repeat Indicator	Defines how many times the information should be sent to the user.

The route call results or event report result include the parameters shown in table 4 and table 10 respectively,.

Table 4: Route call results parameters

Response Requested or Event Reports Requested	These include: Progress, Routing Success, Answer, Refused Busy, No Answer, Disconnect, Redirected, Routing Failure and Call Ended.

Regulatory considerations:
None identified.

6.3.2 Routing of an originating or incoming call from the PTN to the SP

On defined criteria, the service provider needs an originating or incoming call from a service provider service user to be routed from the PTN to the SP, e.g. based on the dialled digits or CLI in accordance with [1] clause 5.3.2 and [2] clause 5.2.2.

Information flows:

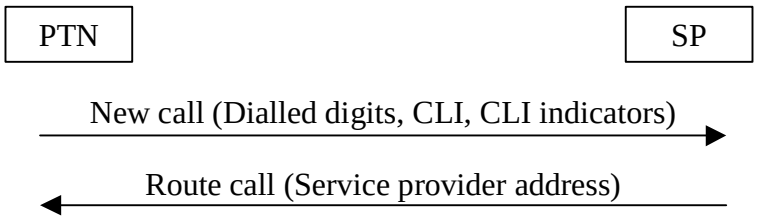


Figure 14: Routing of an originating or incoming call from the PTN to the SP

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification ().

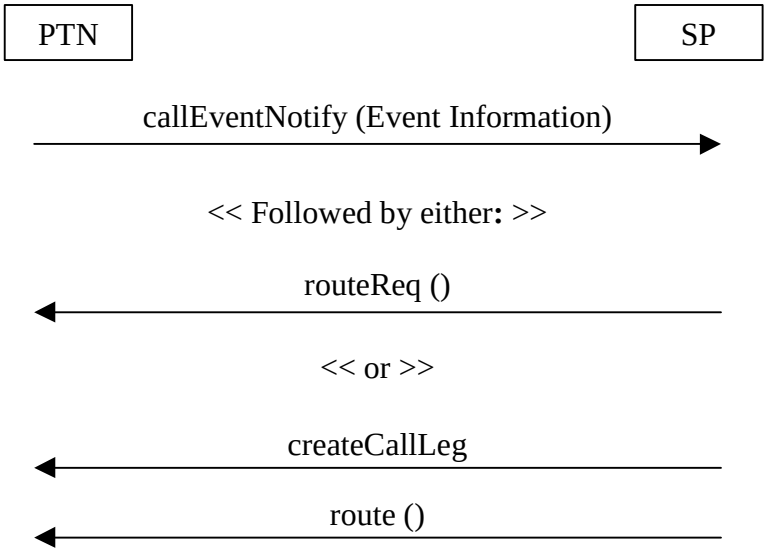


Figure 15: Routing of an originating or incoming call from the PTN to the SP

The Event Information includes the parameters shown in table 1.

The routeReq () and createCallLeg () methods include a Target Address parameter, which is set to route the call to the service provider. The address parameters include the elements shown in table 2.

Regulatory considerations:

National regulatory authorities may impose restrictions on the delivery of some CLI and redirecting address information, therefore the PTNO requires the ability to prevent the forwarding of restricted information.

6.3.3 Indication of an originating or incoming call from the PTN to the SP

On defined criteria, the service provider needs from the PTN the indication of an originating or incoming call from an service provider service user e.g. based on the dialled digits or CLI in accordance with [1] clause 5.3.3 and [2] clause 5.2.2.

Information flows:

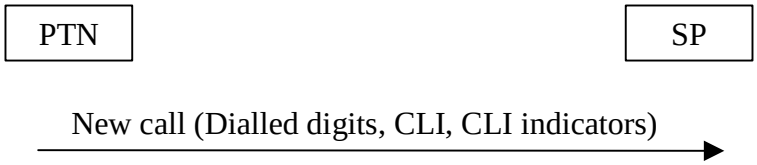


Figure 16: Indication of an originating or incoming call from the PTN to the SP

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification ().

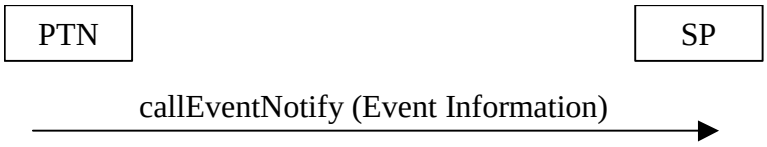


Figure 17: Indication of an originating or incoming call from the PTN to the SP

The Event Information includes the parameters shown in table 1.

Regulatory considerations:

National regulatory authorities may impose restrictions on the delivery of some CLI and redirecting address information, therefore the PTNO requires the ability to prevent the forwarding of restricted information.

6.3.4 Routing of a terminating call from the PTN to the SP

On defined criteria, the service provider needs to receive a terminating call of the service provider's service user, e.g. based on the dialled digits or CLI. After that, the service provider can take a further action to direct the destination and routing of the call in accordance with [1] clause 5.3.4 and [2] clause 5.2.2.

Information flows:

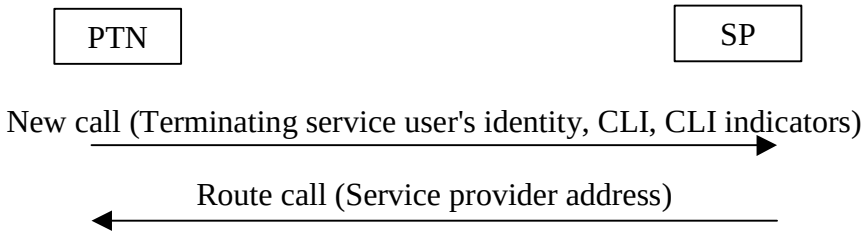


Figure 18: Routing of a terminating call from the PTN to the SP

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification ().

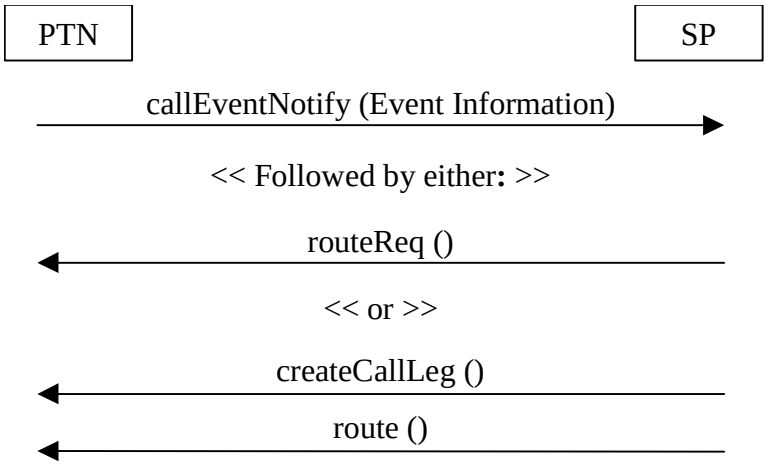


Figure 19: Routing of a terminating call from the PTN to the SP

The Event Information includes the parameters shown in table 1.

The routeReq () and createCallLeg () methods include a Target Address parameter, which is set to route the call to the service provider. The address parameter includes the elements shown in table 2.

Regulatory considerations:

National regulatory authorities may impose restrictions on the delivery of some CLI and redirecting address information, therefore the PTNO requires the ability to prevent the forwarding of restricted information.

6.3.5 Indication of a terminating call from the PTN to the SP

On defined criteria, the service provider needs to receive from the PTN the indication of a terminating call to the service provider's service user, e.g. based on the dialled digits or CLI in accordance with [1] clause 5.3.5and [2] clause 5.2.2.

Information flows:

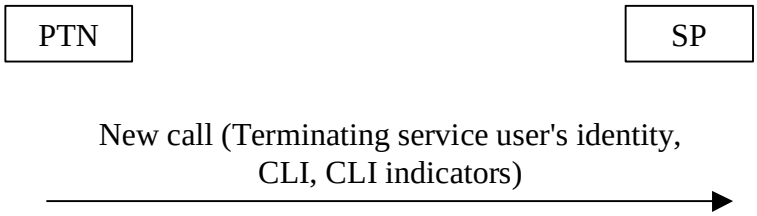


Figure 20: Indication of a terminating call from the PTN to the SP

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification ().

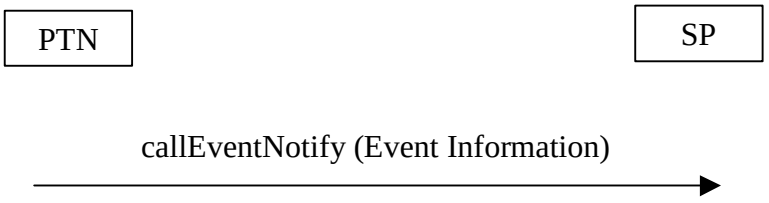


Figure 21: Indication of a terminating call from the PTN to the SP

The Event Information includes the parameters shown in table 1.

Regulatory considerations:

National regulatory authorities may impose restrictions on the delivery of some CLI and redirecting address information, therefore the PTNO requires the ability to prevent the forwarding of restricted information.

6.3.6 Reception of an indication of the cause of an unsuccessful call

On defined criteria and in accordance with [1] clause 5.3.6, the service provider needs to receive from the terminating PTN the indication of an unsuccessful call and the cause value, i.e. either when an indication other than 'alerting tone' is returned to the calling party, or when a 'no reply' situation occurs. After receiving this indication, the SP needs the ability to send a response to the PTN for further control of the call. The indication is needed on the basis of defined call unsuccessful cause values, such as 'called party busy' or 'no reply'. There must be a mechanism for the support of charging between the PTNO and the service provider.

Information flows:

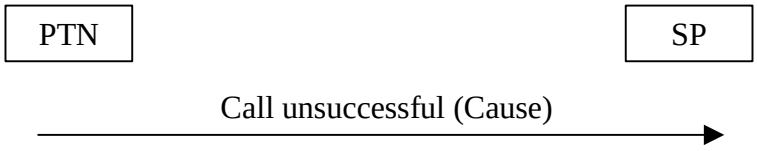


Figure 22: Reception of an indication of the cause of an unsuccessful call

API mapping example:

Pre-conditions: Call routeing has been attempted using routeReq (responseRequested) or createCallLeg (), eventReportReq (eventReportsRequested) and route ().

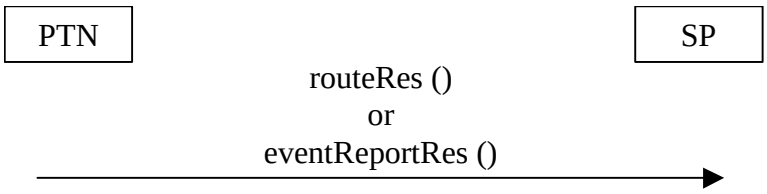


Figure 23: Reception of an indication of the cause of an unsuccessful call

All the above methods include the parameter shown in table 5.

Table 5: Results parameter

Event Report	Refused Busy, No Answer, Disconnect, Redirected, Routing Failure and Call Ended.

Regulatory considerations:

None identified.

6.3.7 Provision of information for the destination and routing of a call

The service provider needs the ability to provide call destination and routing information. If the service provider is not connected to the speech path, the service provider needs to receive the indication of a call in accordance with [1] clause 5.3.7. The service provider also needs the ability to return a message to the PTN for controlling the destination and routing of the call. There must be a mechanism for the support of charging between the PTNO and the service provider.

Information flows:

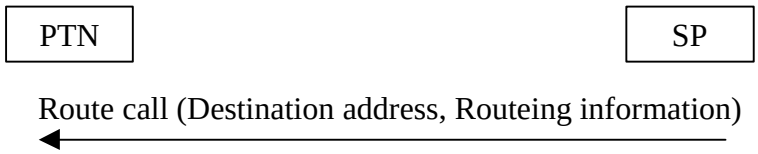


Figure 24: Provision of information for the destination and routing of a call

API mapping example:

Pre-conditions: A call object exists and the service provider has received a callEventNotify (Event Information). If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

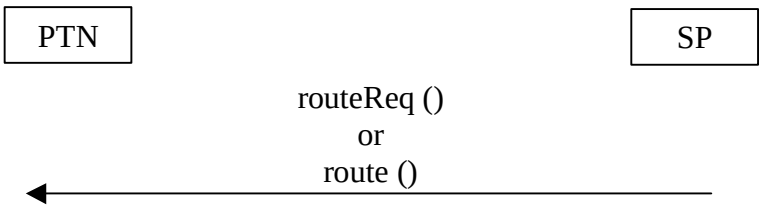


Figure 25: Provision of information for the destination and routing of a call

The routeReq () method includes the parameters shown in table 6.

Table 6: Route request parameters.

Application Information	Alerting Mechanism, Network Access Type, Interworking Indicators, Teleservice, Bearer Service, Party Category, Presentation Address (see table 2) and Generic Information.
Target Address	See table 2.
Originating Address	See table 2.

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission).

Regulatory considerations:

None identified.

6.3.8 Call drop-back

The service provider needs the ability to drop a call back to the same PTN that delivered the call in accordance with [1] clause 5.3.8.

Information flows:

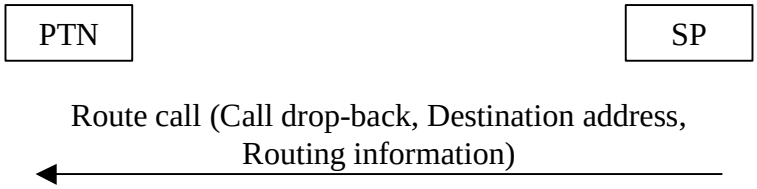
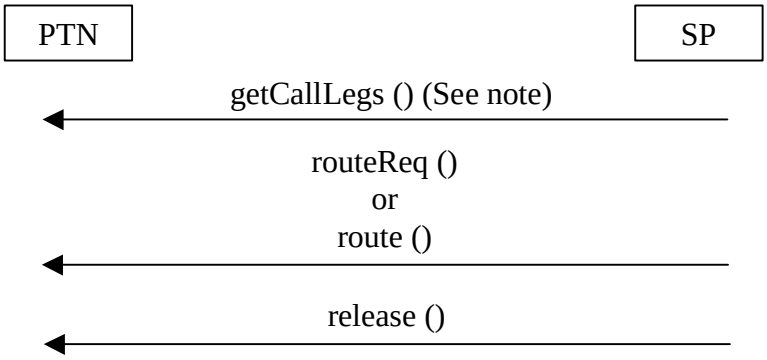


Figure 26: Call drop-back

API mapping example:

Pre-conditions: A call has been routed to the service provider and call drop-back is supported by the underlying PTN. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.



NOTE: The getCallLegs () method returns the identities of the call legs to the service provider.

Figure 27: Call drop-back

The routing methods route () or routeReq () include the parameters shown in table 6.

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission).

Regulatory considerations:

None identified.

6.3.9 User interaction without service charging of the end user

The service provider needs the ability to interact with the end user before any service charging begins. The end user is not charged in accordance with [1] clause 5.3.9. There must be a mechanism for the support of charging between the PTNO and the service provider.

Information flows:

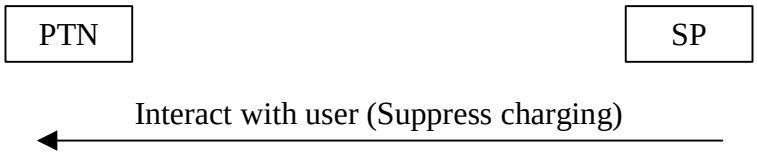


Figure 28: User interaction without service charging of the end user

API mapping example:

Two scenarios are presented, one for calls received by a service provider and one for calls initiated by a service provider. These scenarios assume that there is a network resource capable of supporting the user interaction.

Scenario 1: Call received by service provider.

Pre-conditions: A call object exists and the service provider has received a callEventNotify (Event Information).

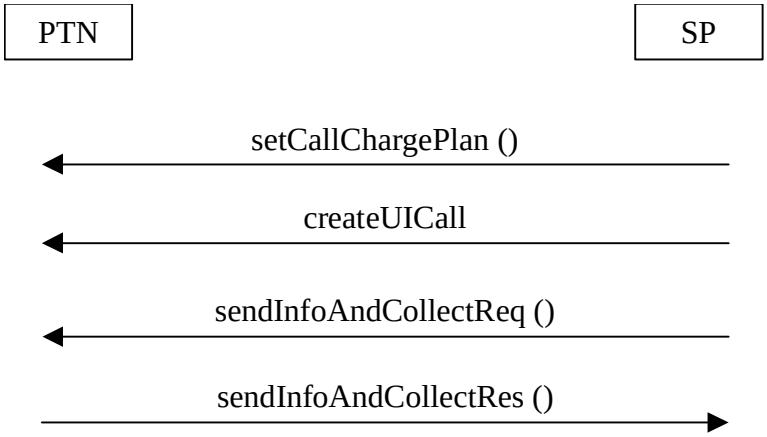


Figure 29: User interaction without service charging of the end user – Scenario 1

The sendInfoAndCollectReq () includes the parameters shown in table 7.

Table 7: sendInfoAndCollectReq () parameters

Information	Specifies the identity of the information to send to the user.
Variable Information	Defines the variable part of the information to send to the user.
Criteria	Specifies additional properties for the collection of information, e.g. minimum length, maximum length, end sequence, start timeout (duration), inter-character time-out.

The sendInfoAndCollectRes () includes the parameters shown in table 8.

Table 8: sendInfoAndCollectRes () parameters

Response	Specifies the type of response from the user, e.g. announcement ended, legal input, no input, timeout, message stored, message not stored.
Information	Specifies the information collected from the user.

The setCallChargePlan () method includes the parameter shown in table 9.

Table 9: setCallChargePlan () parameter

Call Charge Plan	Set up by bilateral agreements between service providers and PTNOs.

Scenario 2: Call initiated by service provider.

Pre-conditions: None.

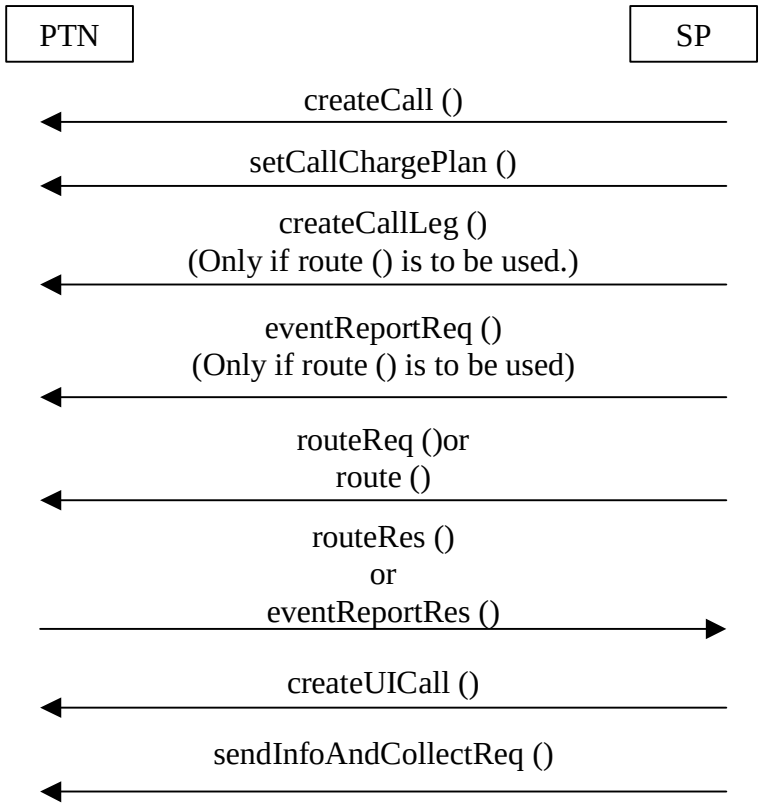


Figure 30: User interaction without service charging of the end user – Scenario 2

The routeRes () and eventReportRes () include the parameter shown in table 10.

Table 10: routeRes () and eventReportRes () parameter

Event Report	E.g. Call Answered.

The sendInfoAndCollectReq () includes the parameters shown in table 11.

Table 11: sendInfoAndCollectReq () parameters

Information	Specifies the identity of the information to send to the user.
Variable Information	Defines the variable part of the information to send to the user.
Criteria	Specifies additional properties for the collection of information, e.g. number of characters.

The sendInfoAndCollectRes () includes the parameters shown in table 12.

Table 12: sendInfoAndCollectRes () parameters

Response	Specifies the type of response from the user.
Information	Specifies the information collected from the user.

The setCallChargePlan () method includes the parameter shown in table 13.

Table 13: setCallChargePlan () parameter

Call Charge Plan	Set up by bilateral agreements between service providers and PTNOs.

Regulatory considerations:

None identified.

6.3.10 Reception of the originally dialled digits

The service provider and PTNO need the ability to receive all originally dialled digits from the other party in accordance with [1] clause 5.3.10 and [2] clause 5.3.1.

Information flows:

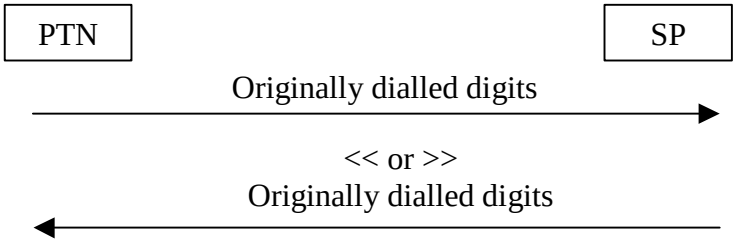


Figure 31: Reception of the originally dialled digits

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotification () and originally dialled digits are available to the PTN.

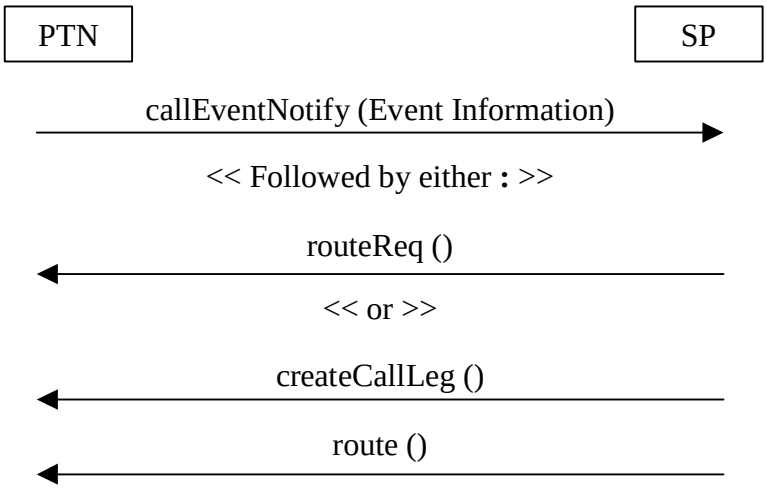


Figure 32: Reception of the originally dialled digits

The Event Information includes the parameters shown in table 1.

Regulatory considerations:

None identified.

6.3.11 Disconnection of a call in progress

The service provider needs the ability to disconnect a call in progress, if either the calling or called party is a service user of the service provider in accordance with [1] clause 5.3.11.

Information flows:

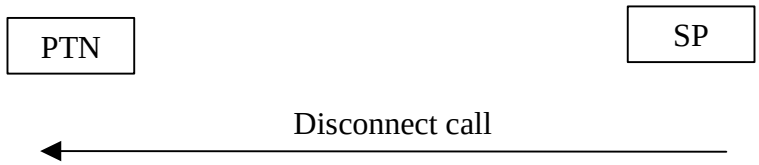


Figure 33: Disconnection of a call in progress

API mapping example:

Pre-conditions: A call exists.

Figure 34: Disconnection of a call in progress

The releaseCall () method includes the parameters shown in table 14.

Table 14: releaseCall () parameters

Call Session Identity or Call Leg Session Identity	Identifies call or call leg to be released.
Cause	Reason for release.

Regulatory considerations:

None identified.

6.3.12 Connection of a call to an interactive voice response unit in the PTN

The service provider needs the ability to instruct the PTN to connect a call to an interactive voice response (IVR) unit in the PTN in accordance with [1] clause 5.3.12.

Information flows:

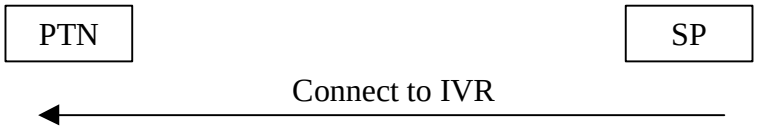


Figure 35: Connection of a call to an interactive voice response unit in the PTN

API mapping example:

Pre-conditions: A call object exists and the service provider has received a callEventNotify (Event Information), or the service provider has initiated a call using either routeReq () or createCallLeg () and route (). A suitable IVR is available in the PTN.

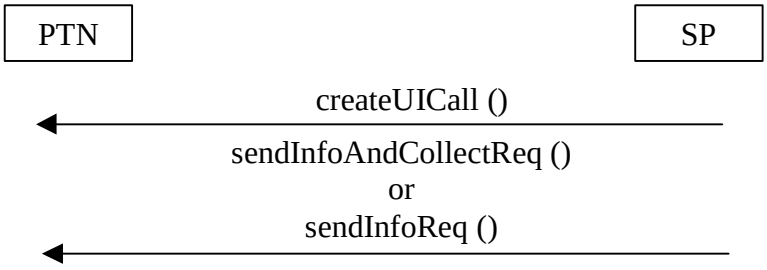


Figure 36: Connection of a call to an interactive voice response unit in the PTN

The sendInfoAndCollectReq () method includes the parameters shown in table 15.

Table 15: sendInfoAndCollectReq () parameters

Information	Specifies the identity of the information to send to the user.
Variable Information	Defines the variable part of the information to send to the user.
Criteria	Specifies additional properties for the collection of information, e.g. minimum length, maximum length, end sequence, start timeout (duration), inter-character time-out.

The sendInfoReq () method includes the parameters shown in table 16.

Table 16: sendInfoReq () parameters

Information	Specifies the identity of the information to send to the user.
Variable Information	Defines the variable part of the information to send to the user.
Repeat Indicator	Defines how many times the information should be sent to the user.

Regulatory considerations:

None identified.

6.3.13 Alternate routing of a call or the indication of a call to another 'point of presence' of the SP

The service provider and PTN need the ability to instruct the other party to route calls or the indications of calls to another 'point of presence', if the first 'point of presence' is not reachable in accordance with [1] clause 5.3.13 and [2] clause 5.3.2.

Information flows:

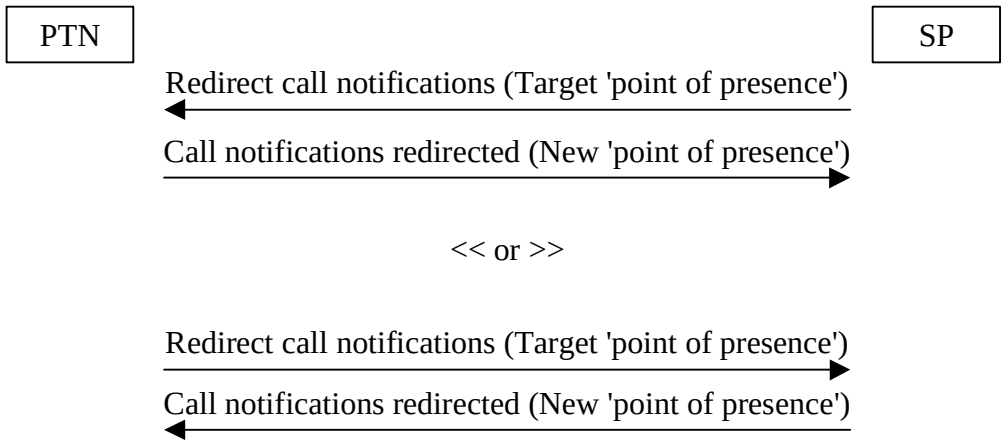


Figure 37: Alternate routing of a call or the indication of a call to another 'point of presence' of the SP

API mapping example:

Pre-conditions: Call event notifications have been enabled using enableCallNotifications (). Calls may be in progress, but these will not be affected. 'Multiple points of presence' are supported by the PTN and the conditions of their use have been agreed between the SP and a PTNO.

Two scenarios are shown, one showing initiation by the service provider, and one showing initiation by the PTN.

Scenario 1: Initiation by the service provider.

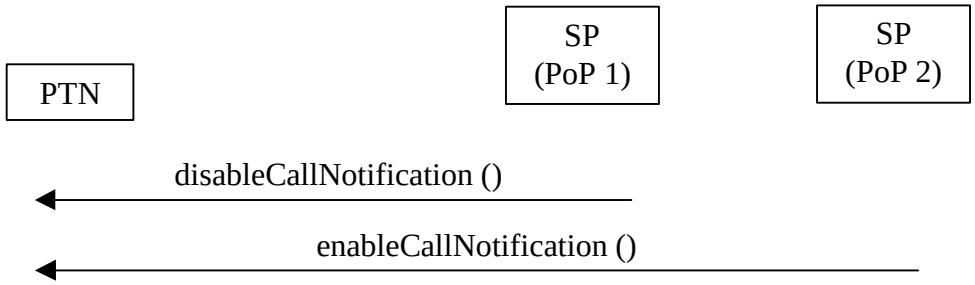


Figure 38: Alternate routing of a call or the indication of a call to another 'point of presence' of the SP

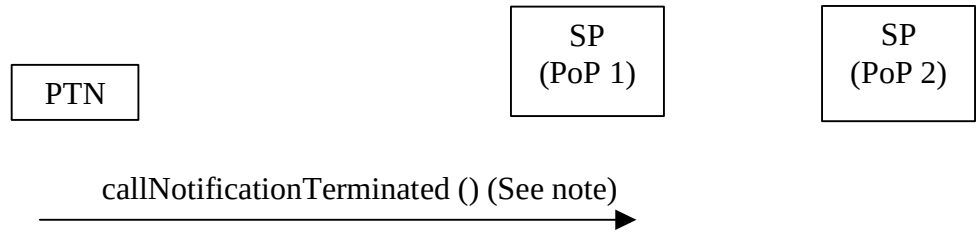
The above methods include the parameter shown in table 17.

Table 17: Call notification parameter

Event Criteria	Includes Destination Address and Origination Address, which identify the address or address range to which the call event notifications relate.

NOTE: The order of the above flows could be reversed to prevent an interruption of service if a suitable bilateral agreement between a service provider and PTNO exists. The requirements for multiple points of presence need to be defined further, e.g. they could be used for load sharing, worker standby, or different services.

Scenario 2: Initiation by the PTN.



NOTE: There are no parameters associated with the callNotificationTerminated () method, therefore the address of SP PoP 2 cannot be passed to SP PoP 1 as a parameter using this method.

Figure 39: Alternate routing of a call or the indication of a call to another 'point of presence' of the SP

Regulatory considerations:

None identified.

6.4 Supplementary call and data processing capabilities

6.4.1 Interrogation of a network termination point for data delivery

The service provider needs the ability to send data to and to receive data from the NTP of a service user without an alerting signal in accordance with [1] clause 5.4.1. There must be a mechanism for the support of charging between the PTNO and the service provider.

Information flows:

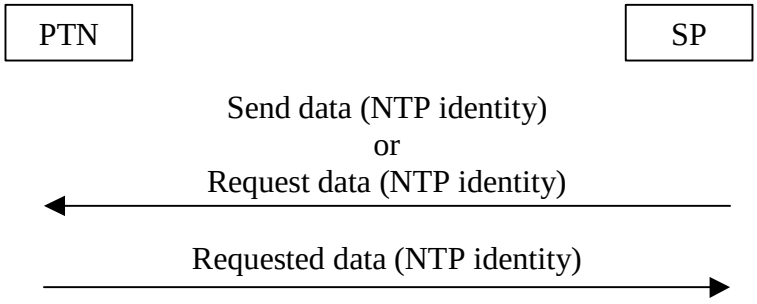


Figure 40: Interrogation of a network termination point for data delivery

API mapping example:

There is no explicit support for interrogating customer premises equipment, but two possible scenarios are shown, one using call routing and one using messaging.

Scenario 1: Call routing.

Pre-conditions: None.

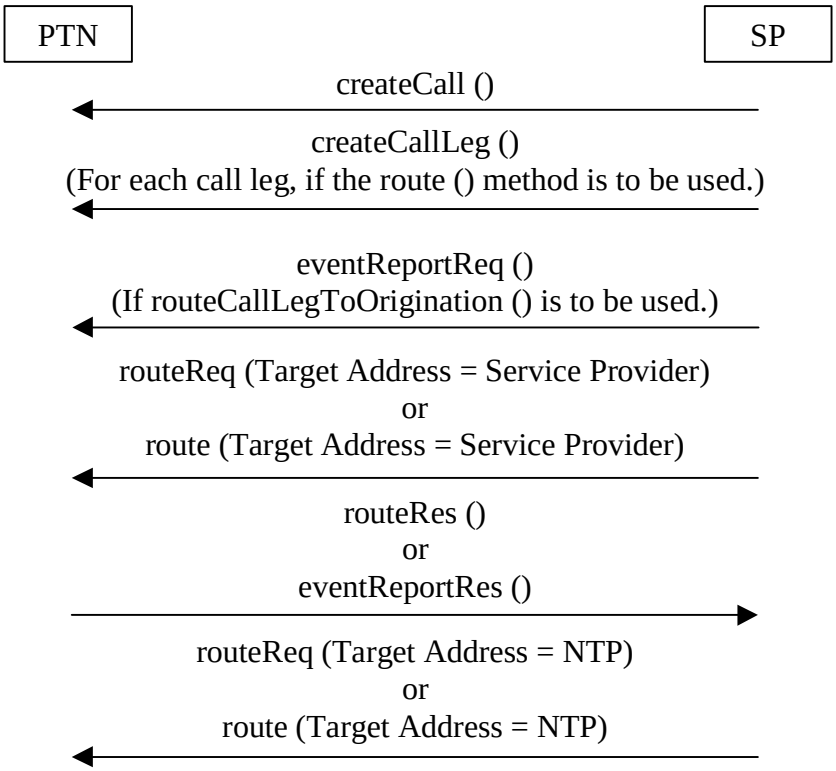


Figure 41: Interrogation of a network termination point for data delivery

The routeReq () and createCallLeg () methods above include the parameters shown in table 6.

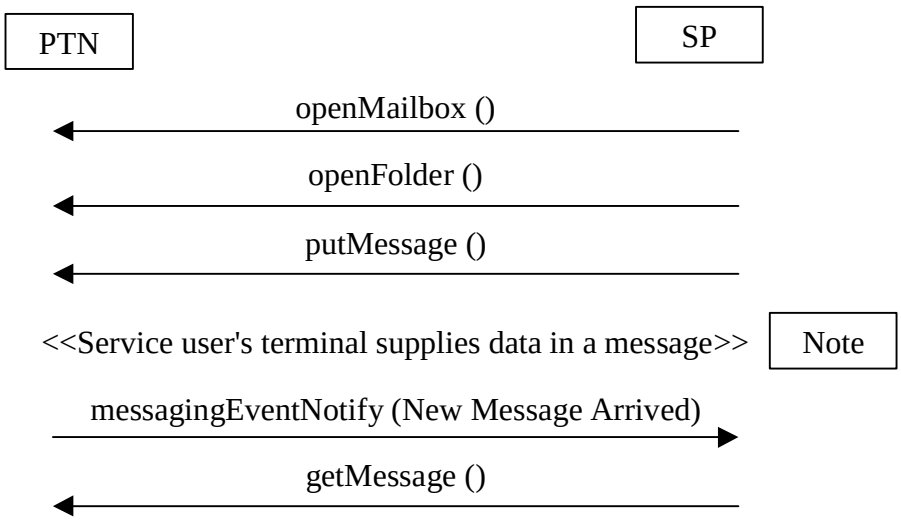
The routeRes () and eventReportRes () include the parameter shown in table 18.

Table 18: Result parameter

Event Report	Progress, Routing Success, Answer, Refused Busy, No Answer, Disconnect, Redirected, Routing Failure and Call Ended.

Scenario 2: Messaging.

Pre-conditions: The service user's terminal is set up as a mailbox, a folder has been created using createFolder (), and message notifications have been enabled using enableMessagingNotification (). The resources to provide messaging must be available in the underlying PTN.



NOTE: It may be necessary to close the mailbox and folder after the putMessage () and re-open them after the messagingEventNotify ().

Figure 42: Interrogation of a network termination point for data delivery

The openMailbox () method includes the parameters shown in table 19.

Table 19: openMailbox () parameters

Mailbox Identity	E.g. Identifies the NTP delivery service.
Authentication Information	Key or password, if required.

The openFolder () method includes the parameter shown in table 20.

Table 20: openFolder () parameter

Folder Identity	E.g. Identifies the NTP.

The putMessage () and getMessage () methods include the parameter shown in table 21.

Table 21: putMessage () and getMessage () parameter

Message	Message supplied by the service provider in the case of putMessage (), or message returned to the service provider in the case of getMessage ().

Regulatory considerations:

Data protection legislation will need to be considered when accessing the service user's data.

6.4.2 Overriding of the 'incoming call barring' supplementary service

The service provider needs the ability to request the PTN to override any 'incoming call barring' supplementary service activated on one of the service provider's service users' lines, provided this is allowed by bilateral agreements between the SP and the provider of the 'incoming call barring' supplementary service in accordance with [1] clause 5.4.2.

Information flows:

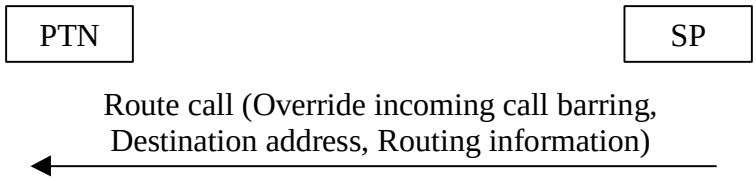


Figure 43: Overriding of the 'incoming call barring' supplementary service

API mapping example:

Pre-conditions: A call object exists and the service provider may have either received a callEventNotify (Event Information), or be initiating a call. Overriding the 'incoming call barring' supplementary service is supported in the underlying PTN. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

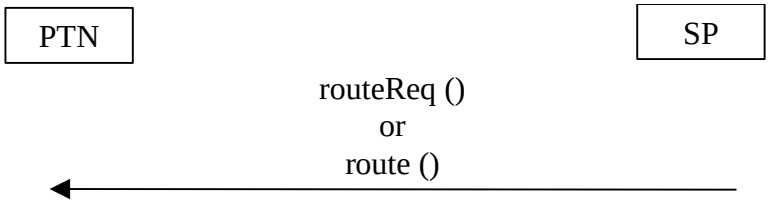


Figure 44: Overriding of the 'incoming call barring' supplementary service

The route () and routeReq () methods include the parameters shown in table 6.

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission). The Interworking Indicators, Party Category and Generic Information (defined as unspecified service-service information) are operator specific and could include an 'override incoming call barring' request, subject to bilateral agreement between a service provider and a PTNO.

Regulatory considerations:

National regulatory authorities may place restrictions on overriding incoming call barring.

6.4.3 Bypassing of the 'call diversion' supplementary service

The SP needs the ability to request the PTN to bypass any 'call diversion' supplementary service on one of the service provider's service users' lines, provided this is allowed by bilateral agreements between the SP and the provider of the 'call diversion' supplementary service in accordance with [1] clause 5.4.3.

Information flows:

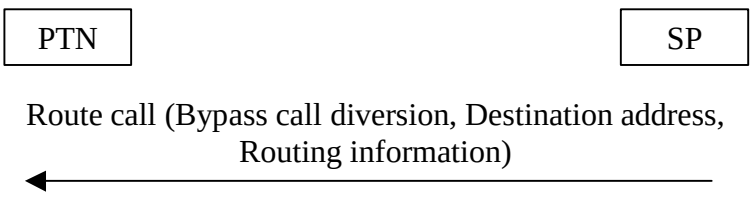


Figure 45: Bypassing of the 'call diversion' supplementary service

API mapping example:

Pre-conditions: A call object exists and the service provider may have either received a callEventNotify (Event Information), or be initiating a call. Bypassing the 'call diversion' supplementary service is supported by the underlying PTN. If the route () method is to be used, a call leg object has been created using createCallLeg (), which includes address parameters containing the elements shown in table 2.

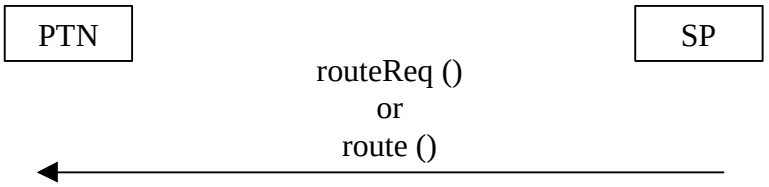


Figure 46: Bypassing of the 'call diversion' supplementary service

All the above methods include the following parameters shown in table 6.

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission). The Interworking Indicators, Party Category and Generic Information (unspecified service-service information) are operator specific and could include a 'bypass call diversion' request, subject to bilateral agreement between a service provider and a PTNO.

Regulatory considerations:

National regulatory authorities may place restrictions on bypassing call diversion.

6.4.4 Message waiting indication

The service provider needs the ability to activate and deactivate a message waiting indication to a service provider service user's line in accordance with [1] clause 5.4.4.

Information flows:

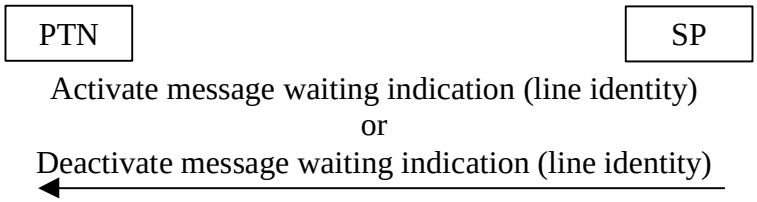


Figure 47: Message waiting indication

API mapping example:

Activation and deactivation of message waiting indication is not explicitly supported by the API, but two scenarios are shown, one using call routing and one using messaging.

Scenario 1: Call routing.

Pre-conditions: A messaging waiting indication control unit has been set up with defined parameters for control the application and removal of messaging waiting indication. The resources to provide message waiting indication must be available in the underlying PTN.

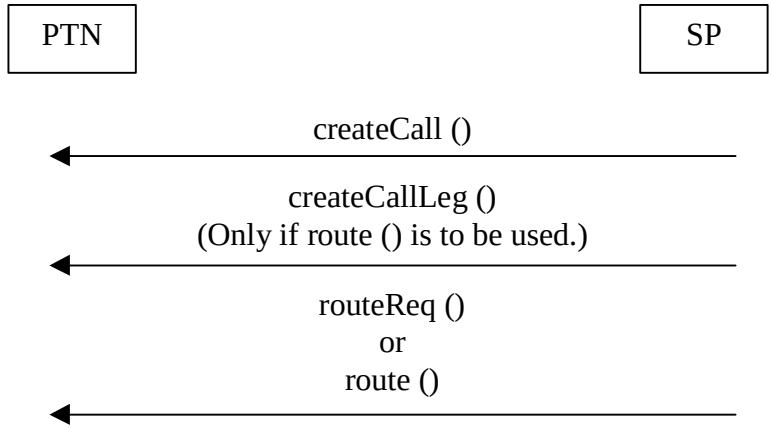


Figure 48: Message waiting indication

The createCallLeg () and routeReq () methods above include the parameters shown in table 6.

The Target Address is used to route the call to the message waiting indication control unit, and the other parameters are used to provide the control information, subject to bilateral agreements between service providers and PTNOs.

The present address parameters do not explicitly contain any routing information, e.g. network operator selection, but a bilateral agreement for an 'Undefined' Address Plan could include routing codes.

Scenario 2: Messaging.

Pre-conditions: A message waiting indication control unit has been set up with a mailbox interface. The resources to provide mailboxes must be available in the underlying PTN.

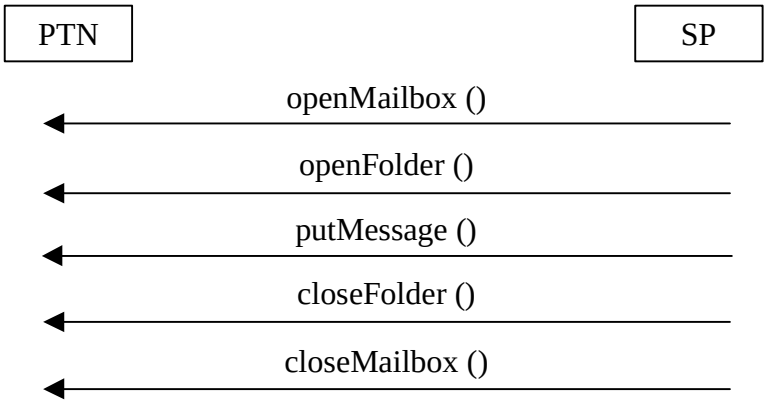


Figure 49: Message waiting indication

The openMailbox () method includes the parameters shown in table 22.

Table 22: openMailbox () parameters

Mailbox Identity	E.g. Identifies the message waiting indication control function.
Authentication Information	Key or password, if required.

The openFolder () method includes the parameter shown in table 23.

Table 23: openFolder () parameter

Folder Identity	E.g. Identifies the service user's line.

The putMessage () method includes the parameter shown in table 24.

Table 24: putMessage () parameter

Message	Message supplied by the service provider containing the message waiting activation or deactivation requests.

Regulatory considerations:

None identified.

6.5 Charging-related capabilities

6.5.1 Changes in the charging rate of a call

The service provider needs the ability to instruct the PTN to change the charging rate of a call, applied to the service provider's service user, during the call set-up phase and the duration of the call in accordance with [1] clause 5.5.1.

Information flows:

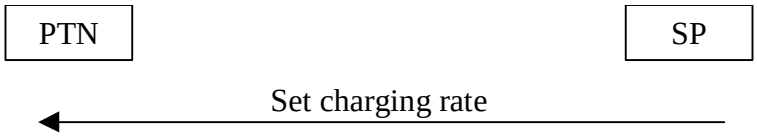


Figure 50: Changes in the charging rate of a call

API mapping example:

The API can only set the call charging plan during the call set-up phase, before routeReq () or route () are invoked. This plan could include pre-determined charging rates for different phases of a call, subject to bilateral agreement between a service provider and a PTNO, e.g. calling user may not be charged for the user interaction phase of a call.

Pre-conditions: A call object exists and the call has not been routed to a destination address. A charging rate change mechanism is available in the underlying PTN.

Figure 51: Changes in the charging rate of a call

The setCallChargePlan () method includes the parameter shown in table 25.

Table 25: setCallChargePlan () parameter

Call Charge Plan	Set up by bilateral agreements between service providers and PTNOs. This may be a complex charge plan to cover all possible call phases.

Regulatory considerations:

National regulatory authorities may impose restrictions on changing the charging rate during a call.

6.6 Traffic-related capabilities

6.6.1 Event traceability

The service provider and PTNO need the ability to perform event traceability with the other party, which should produce relevant call tracing data in accordance with [1] clause 5.6.1 and [2] clause 5.4.1.

Information flows:

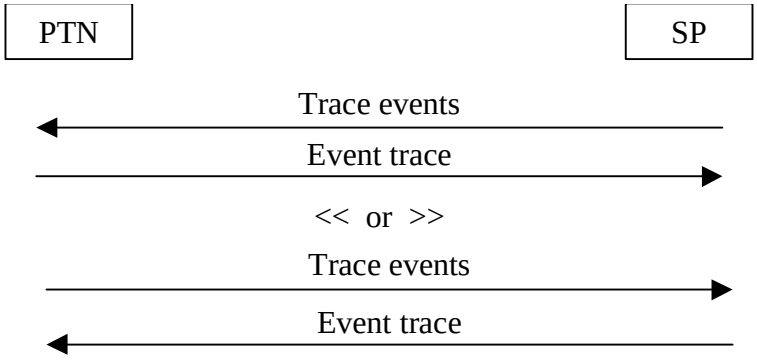


Figure 52: Event traceability

API mapping example:

Four scenarios are given below, showing various call-related and service-related event tracing.

Scenario 1: Call-related events using API Generic Call methods.

Pre-conditions: Call object exists and either the application has received a callEventNotify (Event Information), or the application is about to initiate a call.

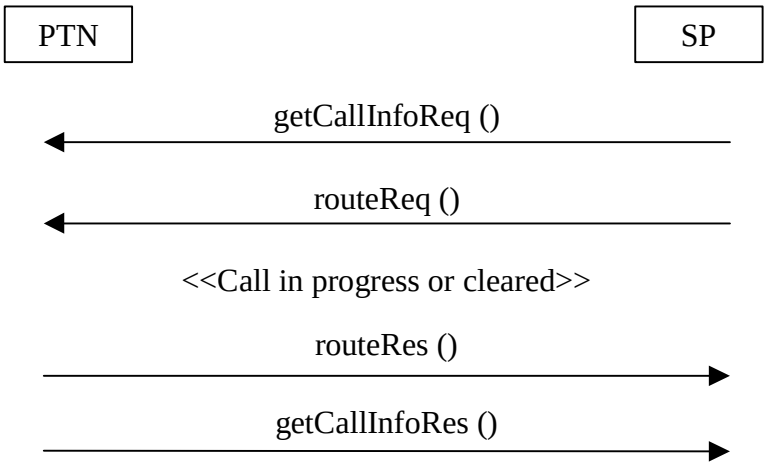


Figure 53: Event traceability

The getCallInfoReq () method includes the parameter shown in table 26.

Table 26: getCallInfoReq () parameter

Call Information Requested	Call Initiation Start Time, Connected To Resource Time, Connected To Destination Time, End Time, Release Cause, Leg Information.

The routeReq () method includes the parameter shown in table 27.

Table 27: Route call request parameter

Response Requested	Call Refused Busy, Call No Answer, Call Answer, Addressed Party Requests Call Release, Call Ended.

The routeRes () method includes the parameter shown in table 28.

Table 28: Route call result parameter

Event Report	Call Refused Busy, Call No Answer, Call Answer, Addressed Party Requests Call Release, Call Ended.

The getCallInfoRes () method includes the parameter shown in table 29.

Table 29: getCallInfoRes () parameter

Call Information Report	Call Initiation Start Time**,** Connected To Resource Time, Connected To Destination Time, End Time, Release Cause, Leg Information.

The address parameters need to contain routing information, including selected network operator, alternate network operator, and chained transit network selection. Actual network codes may be agreed bilaterally or centralized by national regulatory authorities (as harmonized by groups of NRAs or the European Commission).

Scenario 2: Call-related events using API Call Leg methods.

Pre-conditions: Call object exists and the service provider has either received a callEventNotify (Event Information), or is about to initiate a call.

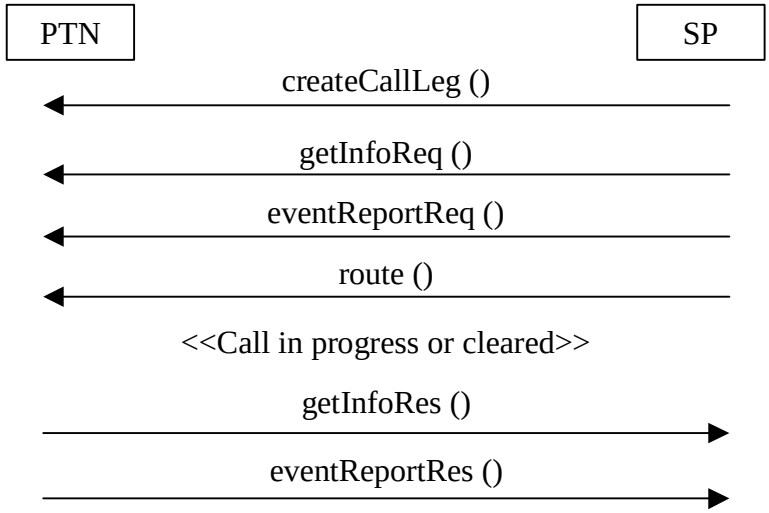


Figure 54: Event traceability

The getInfoReq () method includes the parameter shown in table 30.

Table 30: getInfoReq () parameter

Call Leg Information Requested	Call Initiation Start Time, Connected To Resource Time, Connected To Destination Time, End Time, Release Cause, Leg Information.

The eventReportReq () method includes the parameters shown in table 31.

Table 31: eventReportReq () parameter

Event Reports Requested	Progress, Routing Success, Answer, Refused Busy, No Answer, Disconnect, Redirected, Service Code, Routing Failure and Call Ended.

The getInfoRes () method includes the parameter shown in table 32.

Table 32: getInfoRes () parameter

Call Leg Information Report	Call Initiation Start Time, Connected To Resource Time, Connected To Destination Time, End Time, Release Cause, Leg Information.

The eventReportRes () method includes the parameter shown in table 33.

Table 33: eventReportRes () parameter

Event Report	Progress, Routing Success, Answer, Refused Busy, No Answer, Disconnect, Redirected, Service Code, Routing Failure and Call Ended.

Scenario 3: Service-related and call-related events using Generic Call Control methods.

Pre-conditions: Call object exists and the service provider has either received a callEventNotify (Event Information), or is about to initiate a call. Call overload control has been enabled using setCallLoadControl ().

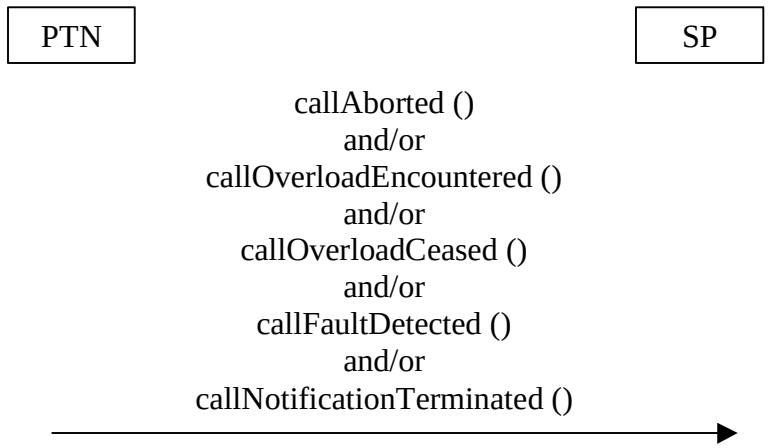


Figure 55: Event traceability

The callAborted () method contains no parameters providing further information.

The callOverloadEncountered () and callOverloadCeased () methods include the parameter shown in table 34.

Table 34: callOverloadEncountered () and callOverloadCeased () parameter

Overload Type	Undefined, inbound calls causing overload, outbound calls causing overload.

The callFaultDetected () method includes the parameter shown in table 35.

Table 35: callFaultDetected () parameter

Fault	Undefined, User aborted, Timeout on release, Timeout on interrupt.

The callNotificationTerminated () method has no parameters.

Scenario 4: Service-related events using Framework Integrity Management methods.

Pre-conditions: A fault statistics recording mechanism is available in the PTN.

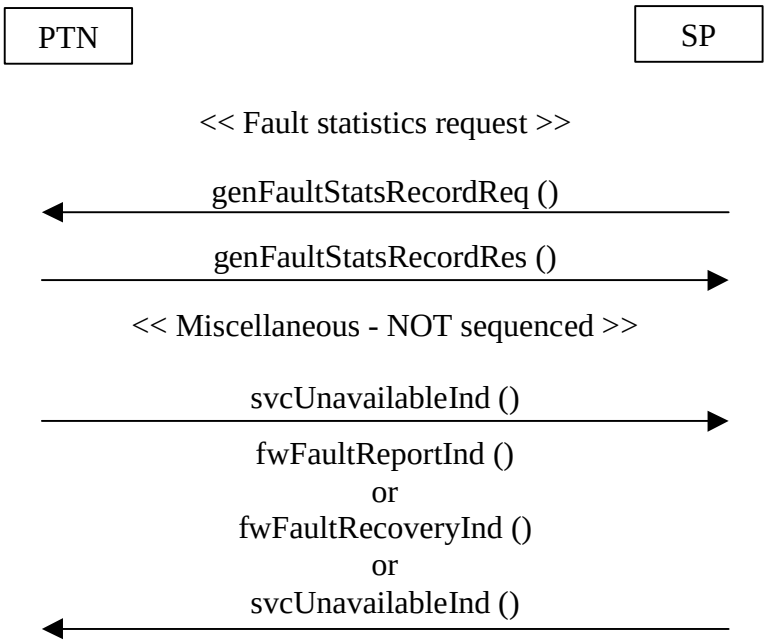


Figure 56: Event traceability

The genFaultStatsRecordReq () method includes the parameter shown in table 36.

Table 36: genFaultStatsRecordReq () parameter

Time Period	Period over which fault statistics are to be generated.
Service Identity List	Lists the services for which the statistics are required.

The scvUnavailableInd () methods include the parameters shown in table 37.

Table 37: scvUnavailableInd () parameter

Service Identity	Identifies the service which is unavailable.
Reason	Undefined, Local failure, Gateway failure, Overloaded, Closed.

The fwFaultReportInd () and fwFaultRecoveryInd () methods include the parameter shown in table 38.

Table 38: fwFaultReportInd () and fwFaultRecoveryInd () parameter

Fault	Undefined, Local failure, Gateway failure, Protocol error.

Regulatory considerations:

None identified.

6.6.2 Traffic control capabilities

The service provider and PTNO need the ability to control the signalling and/or call traffic between the service provider's equipment and the PTN's equipment in accordance with [1] clause 5.6.2 and [2] clause 5.4.2.

Information flows:

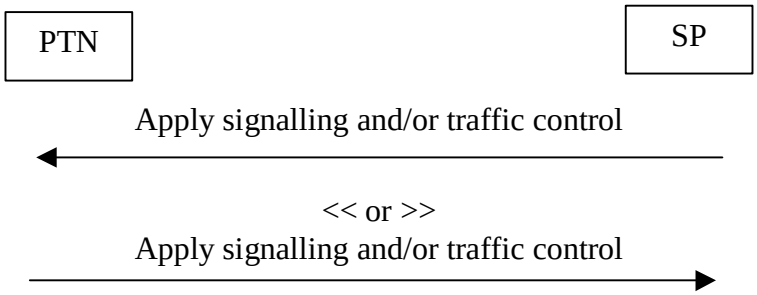


Figure 57: Traffic control capabilities

API mapping example:

Two scenarios are shown, one service provider initiated and one network operator initiated.

Scenario 1: Service provider initiated.

Pre-conditions: Call notifications enabled and a load control mechanism is available in the PTN.

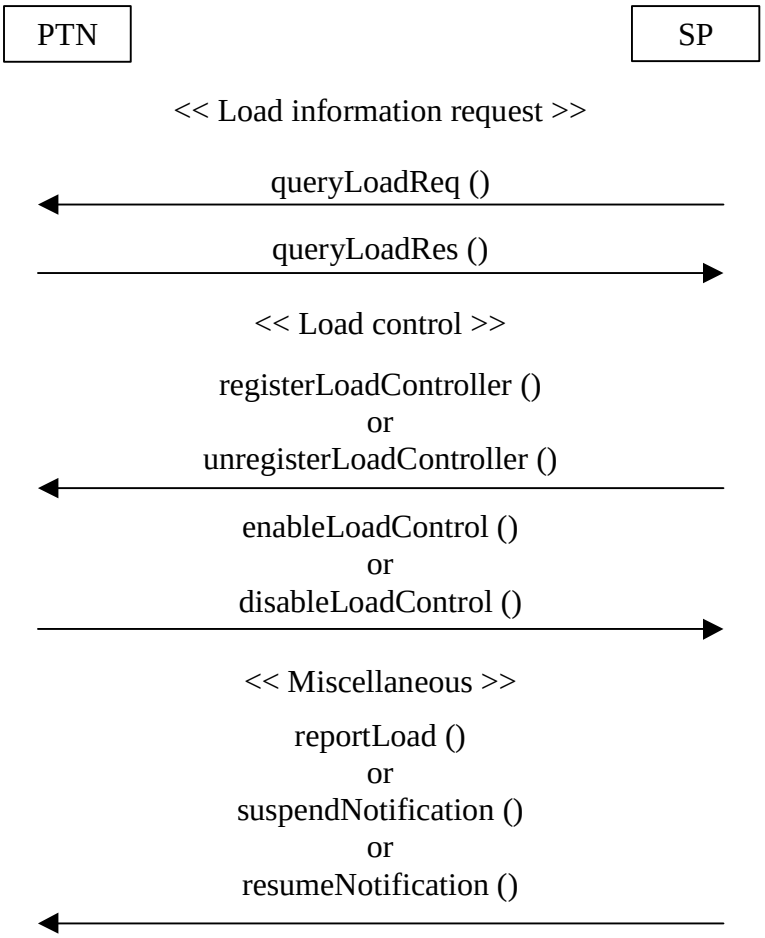


Figure 58: Traffic control capabilities

The queryLoadReq () method includes the parameters shown in table 39.

Table 39: queryLoadReq () parameters

Service Identities	Identities of the services for which statistics are required.
Time Interval	Interval within which the load statistics are generated.

The queryLoadRes () method includes the parameter shown in table 40.

Table 40: queryLoadRes () parameter

Load Statistics	Specifies the framework supplied statistics.

The registerLoadController (), unregisterLoadController () and disableLoadControl () methods include the parameter shown in table 41.

Table 41: Load control parameter

Service Identities	Identities of the services for which statistics are required.

The enableLoadControl () method includes the parameter shown in table 42.

Table 42: enableLoadControl () parameter

Load Statistics	Service identities, Load value, Load level, Time stamp.

The reportLoad () method includes the parameter shown in table 43.

Table 43: reportLoad () parameter

Load Level	Specifies the application's load level.

The suspendNotification () and resumeNotification () methods include the parameter shown in table 44.

Table 44: Notification parameter

Service Identities	Identities of the services for which load notifications should be suspended or resumed.

Scenario 2: Network operator initiated.

Pre-conditions: Call notifications may be enabled and a load control mechanism is available in the PTN. The load controller has been registered using registerLoadControl (), and call overload control has been enabled using setCallLoadControl ().

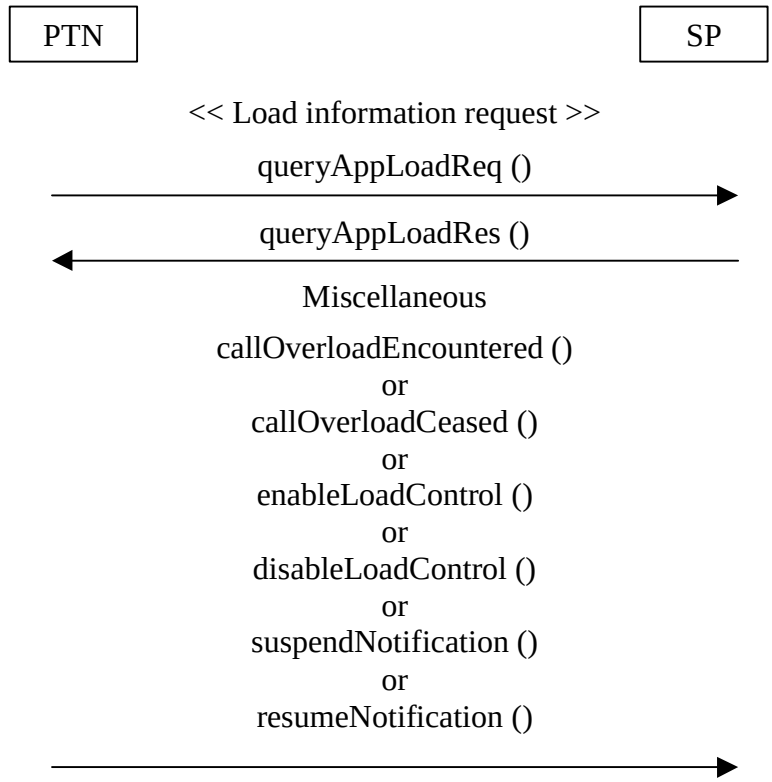


Figure 59: Traffic control capabilities

The queryAppLoadReq () method includes the parameters shown in table 45.

Table 45: queryAppLoadReq () parameters

Service Identities	Identities of the services for which statistics are required.
Time Interval	Interval within which the load statistics are generated.

The queryAppLoadRes () method includes the parameter shown in table 46.

Table 46: queryAppLoadRes () parameter

Load Statistics	Specifies the framework supplied statistics.

The callOverloadEncountered () and callOverloadCeased () methods include the parameter shown in table 47.

Table 47: Overload parameter

Overload Type	Undefined, Inbound calls, Outbound calls.

The enableLoadControl () method includes the parameter shown in table 48.

Table 48: enableLoadControl () parameter

Load Statistics	Specifies the new load statistics.

The disableLoadControl () method includes the parameter shown in table 49.

Table 49: disableLoadControl () parameter

Service Identities	Identities of the services for which the load has changed to normal.

The suspendNotification () and resumeNotification () methods do not contain any parameters.

Regulatory considerations:

None identified.

6.6.3 Avoidance of the cyclical routing of a call

The service provider and PTNO need the ability to detect and stop cyclical routing of a call between the PTN and the service provider's equipment in accordance with [1] clause 5.6.3 and [2] clause 5.4.3.

Information flows:

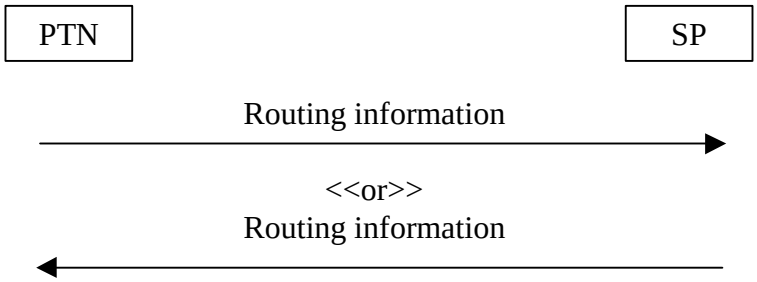


Figure 60: Avoidance of the cyclical routing of a call

API mapping example:

Two scenarios are shown, one for information provided with a new call, and one for information provided during the call set-up phase.

Scenario 1: New call phase.

Pre-conditions: Call event notifications have been enabled using enableCallNotifications () and the parameter information is available in the PTN.

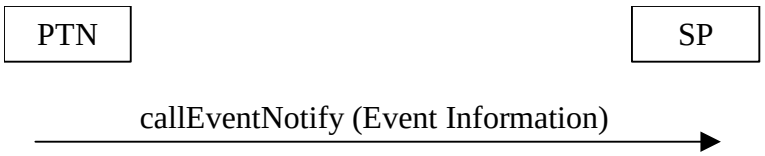


Figure 61: Avoidance of the cyclical routing of a call

The Event Information includes the parameters shown in table 1,which can be used to help prevent cyclical routing. 'Hop' counters, as suggested in the network operators' requirements [2], are not defined in the API methods. Suitable counters could be included, subject to suitable bilateral agreements between service providers and network operators, as Interworking Indicators or Generic Information.

Scenario 2: Call set-up phase.

Pre-conditions: An interrupt-type event has been set for redirected calls using eventReportReq () and the call has been routed using createCallLeg () and route (). The parameter information is available in the PTN.

Figure 62: Avoidance of the cyclical routing of a call

The eventReportRes () method includes the parameter shown in table 50.

Table 50: eventReportRes () parameter

Event Report	Undefined, Progress, Routing success, Answer, Refused busy, No answer, Disconnect, Redirected (see below), Service code, Routing failure, Call ended.

The Redirected element is a forward address which can be used by the service provider to help detect cyclical routing. If cyclical routing is detected, the service provider can disconnect the call leg using the release () method and choose an alternative call treatment.

Regulatory considerations:
None identified.

Annex A (informative): Bibliography

- ETSI ES 201 915 (all parts): "Open Service Access; Application Programming Interface".

History

Document history		
V1.1.1	February 2001	Membership Approval Procedure MV 20010413: 2001-02-13 to 2001-04-13
V1.1.1	May 2001	Publication