

ETSI ES 204 915-12 V1.1.1 (2008-05)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Part 12: Charging SCF (Parlay 6)



iTeh STANDARD PREVIEW
(standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/90b6a1fe-ee4d-482d-a5e5-9a471505742f/etsi-es-204-915-12-v1.1.1-2008-05>



Reference

DES/TISPAN-01032-12-OSA

Keywords

API, IDL, OSA, 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

Important notice

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

http://portal.etsi.org/chaicor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.

© The Parlay Group 2008.

All rights reserved.

DECTTM, **PLUGTESTS**TM, **UMTS**TM, **TIPHON**TM, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

Intellectual Property Rights	7
Foreword.....	7
1 Scope	8
2 References	8
3 Definitions and abbreviations.....	8
3.1 Definitions	8
3.2 Abbreviations	8
4 Charging SCF.....	9
4.1 General requirements on support of methods.....	9
5 Sequence Diagrams	9
5.1 Reservation / payment in parts	9
5.2 Immediate Charge	11
6 Class Diagrams.....	13
7 The Service Interface Specifications	14
7.1 Interface Specification Format	14
7.1.1 Interface Class	15
7.1.2 Method descriptions.....	15
7.1.3 Parameter descriptions.....	15
7.1.4 State Model.....	15
7.2 Base Interface.....	15
7.2.1 Interface Class IpInterface	15
7.3 Service Interfaces	15
7.3.1 Overview	15
7.4 Generic Service Interface	16
7.4.1 Interface Class IpService	16
7.4.1.1 Method setCallback().....	16
7.4.1.2 Method setCallbackWithSessionID().....	16
8 Charging Interface Classes.....	17
8.1 Interface Class IpChargingManager.....	17
8.1.1 Method createChargingSession().....	18
8.1.2 Method createSplitChargingSession()	19
8.2 Interface Class IpAppChargingManager	19
8.2.1 Method sessionAborted().....	20
8.2.2 Method abortMultipleChargingSessions().....	20
8.3 Interface Class IpChargingSession.....	20
8.3.1 Method creditAmountReq().....	21
8.3.2 Method creditUnitReq().....	22
8.3.3 Method debitAmountReq().....	23
8.3.4 Method debitUnitReq().....	23
8.3.5 Method directCreditAmountReq().....	24
8.3.6 Method directCreditUnitReq().....	24
8.3.7 Method directDebitAmountReq().....	25
8.3.8 Method directDebitUnitReq()	26
8.3.9 Method extendLifeTimeReq()	26
8.3.10 Method getAmountLeft().....	26
8.3.11 Method getLifeTimeLeft().....	27
8.3.12 Method getUnitLeft().....	27
8.3.13 Method rateReq().....	27
8.3.14 Method release()	28
8.3.15 Method reserveAmountReq()	28
8.3.16 Method reserveUnitReq()	29

8.4	Interface Class IpAppChargingSession	29
8.4.1	Method creditAmountErr()	30
8.4.2	Method creditAmountRes()	31
8.4.3	Method creditUnitErr()	31
8.4.4	Method creditUnitRes()	32
8.4.5	Method debitAmountErr()	32
8.4.6	Method debitAmountRes()	32
8.4.7	Method debitUnitErr()	33
8.4.8	Method debitUnitRes()	33
8.4.9	Method directCreditAmountErr()	33
8.4.10	Method directCreditAmountRes()	34
8.4.11	Method directCreditUnitErr()	34
8.4.12	Method directCreditUnitRes()	34
8.4.13	Method directDebitAmountErr()	35
8.4.14	Method directDebitAmountRes()	35
8.4.15	Method directDebitUnitErr()	36
8.4.16	Method directDebitUnitRes()	36
8.4.17	Method extendLifeTimeErr()	36
8.4.18	Method extendLifeTimeRes()	37
8.4.19	Method rateErr()	37
8.4.20	Method rateRes()	37
8.4.21	Method reserveAmountErr()	37
8.4.22	Method reserveAmountRes()	38
8.4.23	Method reserveUnitErr()	38
8.4.24	Method reserveUnitRes()	38
8.4.25	Method sessionEnded()	39
9	State Transition Diagrams	40
9.1	State Transition Diagrams for IpChargingSession	40
9.1.1	Session Created State	41
9.1.2	Amount Reserved State	41
9.1.3	Volume Reserved State	41
9.1.4	Reservation Ended State	41
10	Content Based Charging Service Properties	41
11	Data Definitions	42
11.1	Charging Data Definitions	42
11.1.1	IpChargingManager	43
11.1.2	IpChargingManagerRef	43
11.1.3	IpAppChargingManager	43
11.1.4	IpAppChargingManagerRef	43
11.1.5	IpChargingSession	43
11.1.6	IpChargingSessionRef	43
11.1.7	IpAppChargingSession	43
11.1.8	IpAppChargingSessionRef	43
11.1.9	TpApplicationDescription	43
11.1.10	TpAppInformationSet	43
11.1.11	TpAppInformation	44
11.1.12	TpAppInformationType	44
11.1.13	TpSessionEndedCause	44
11.1.14	TpMerchantAccountID	44
11.1.15	TpCorrelationID	44
11.1.16	TpCorrelationType	44
11.1.17	TpChargingPrice	45
11.1.18	TpAmount	45
11.1.19	TpChargingParameterSet	45
11.1.20	TpChargingParameter	45
11.1.21	TpChargingParameterID	45
11.1.22	TpChargingParameterValue	45
11.1.23	TpChargingParameterValueType	46
11.1.24	TpVolumeSet	46
11.1.25	TpVolume	46

11.1.26	TpUnitID	46
11.1.27	TpChargingSessionID	46
11.1.28	TpPriceVolumeSet	47
11.1.29	TpPriceVolume	47
11.1.30	TpChargingError	47
12	Exception Classes	47
Annex A (normative):	OMG IDL Description of Charging SCF	48
Annex B (informative):	W3C WSDL Description of Charging SCF	49
Annex C (informative):	Java™ API Description of the Charging SCF	50
Annex D (informative):	Contents of 3GPP OSA R7 Charging	51
Annex E (informative):	Description of Charging for 3GPP2 cdma2000 networks	52
E.1	General Exceptions	52
E.2	Specific Exceptions	52
E.2.1	Clause 1: Scope	52
E.2.2	Clause 2: References	52
E.2.3	Clause 3: Definitions and abbreviations	52
E.2.4	Clause 4: Charging SCF	52
E.2.5	Clause 5: Sequence Diagrams	52
E.2.6	Clause 6: Class Diagrams	52
E.2.7	Clause 7: The Service Interface Specifications	52
E.2.8	Clause 8: Charging Interface Classes	53
E.2.9	Clause 9: State Transition Diagrams	53
E.2.10	Clause 10: Content Based Charging Service Properties	53
E.2.11	Clause 11: Data Definitions	53
E.2.12	Clause 12: Exception Classes	53
E.2.13	Annex A (normative): OMG IDL Description of Charging SCF	53
E.2.14	Annex B (informative): W3C WSDL Description of Charging SCF	53
E.2.15	Annex C (informative): Java™ API Description of the Charging SCF	53
Annex F (informative):	Record of changes	54
F.1	Interfaces	54
F.1.1	New	54
F.1.2	Deprecated	54
F.1.3	Removed	54
F.2	Methods	54
F.2.1	New	54
F.2.2	Deprecated	54
F.2.3	Modified	55
F.2.4	Removed	55
F.3	Data Definitions	55
F.3.1	New	55
F.3.2	Modified	55
F.3.3	Removed	55
F.4	Service Properties	55
F.4.1	New	55
F.4.2	Deprecated	56
F.4.3	Modified	56
F.4.4	Removed	56
F.5	Exceptions	56
F.5.1	New	56
F.5.2	Modified	56
F.5.3	Removed	56

F.6 Others	56
History	57

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Full standard:
<https://standards.iteh.ai/catalog/standards/sist/90b6a1fe-ee4d-482d-a5e5-9a471505742f/etsi-es-204-915-12-v1.1.1-2008-05>

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document is part 12 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 204 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";
- Part 3: "Framework";
- Part 4: "Call Control";
- Part 5: "User Interaction SCF";
- Part 6: "Mobility SCF";
- Part 7: "Terminal Capabilities SCF";
- Part 8: "Data Session Control SCF";
- Part 9: "Generic Messaging SCF";
- Part 10: "Connectivity Manager SCF";
- Part 11: "Account Management SCF";
- Part 12: "Charging SCF";**
- Part 13: "Policy Management SCF";
- Part 14: "Presence and Availability Management SCF";
- Part 15: "Multi-Media Messaging SCF";
- Part 16: "Service Broker SCF".

The present document has been defined jointly between ETSI, The Parlay Group (<http://www.parlay.org>) and the 3GPP, in co-operation with a number of JAIN™ Community (<http://www.java.sun.com/products/jain>) member companies.

The present document forms part of the Parlay 6.0 set of specifications.

The present document is equivalent to 3GPP TS 29.198-12 V7.0.0 (Release 7).

1 Scope

The present document is part 12 of the Stage 3 specification for an Application Programming Interface (API) for Open Service Access (OSA).

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

The present document specifies the Charging Service Capability Feature (SCF) aspects of the interface. All aspects of the Charging SCF are defined here, these being:

- Sequence Diagrams.
- Class Diagrams.
- Interface specification plus detailed method descriptions.
- State Transition diagrams.
- Data Definitions.
- IDL Description of the interfaces.
- WSDL Description of the interfaces.
- Reference to the Java™ API description of the interfaces.

The process by which this task is accomplished is through the use of object modelling techniques described by the Unified Modelling Language (UML).

2 References

The references listed in clause 2 of ES 204 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 204 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 6)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 204 915-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 204 915-1 apply.

4 Charging SCF

The following clauses describe each aspect of the Charging Service Capability Feature (SCF).

The order is as follows:

- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another.
- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data Definitions clause shows a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part ES 204 915-2.

4.1 General requirements on support of methods

An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method.

Where a method is not supported by an implementation of a Service interface, the exception P_METHOD_NOT_SUPPORTED shall be returned to any call of that method.

Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.

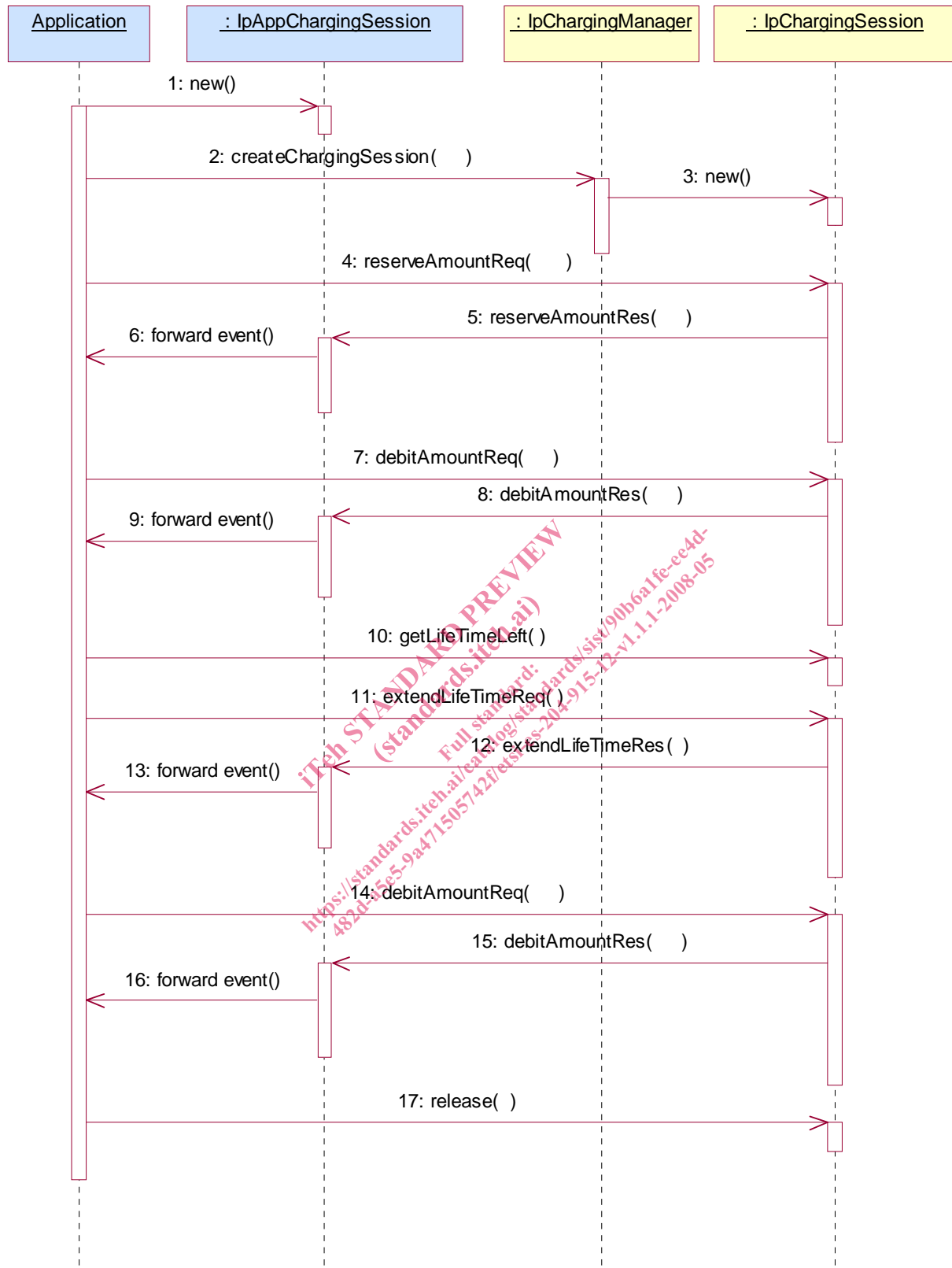
5 Sequence Diagrams

5.1 Reservation / payment in parts

The sequence diagram illustrates how to request a reservation and how to charge a user from the reserved amount, for instance to charge a user for a streamed video which lasts 10 minutes and costs a total of \$2.00. The operations and interfaces that do not provide rating are employed throughout this sequence diagram.

We assume the application has already discovered the Charging SCF. As a result, the application received an object reference pointing to an object that implements the IpChargingManager interface.

The operations which handle units are used exactly the same, except that the amount of application usage is indicated instead of a price.



- 1: The application creates a local object implementing the IpAppChargingSession interface. This object will receive response messages from the IpChargingSession object.
- 2: The application opens a charging session, a reference to a new or existing object implementing IpChargingSession is returned together with a unique session ID.

- 3: In this case a new object is used.
- 4: The application requests the reservation of \$2.00.
- 5: Assuming the criteria for requesting a reservation are met (the application provider has permission to charge the requested amount, the charged user has agreed to pay the requested amount), the amount is reserved in the session. At this point, the application provider knows that the network operator will accept later debit requests up to the reserved amount. So, the application may start serving the user, for instance by sending the video stream.
- 6: The successful reservation is reported back to the application.

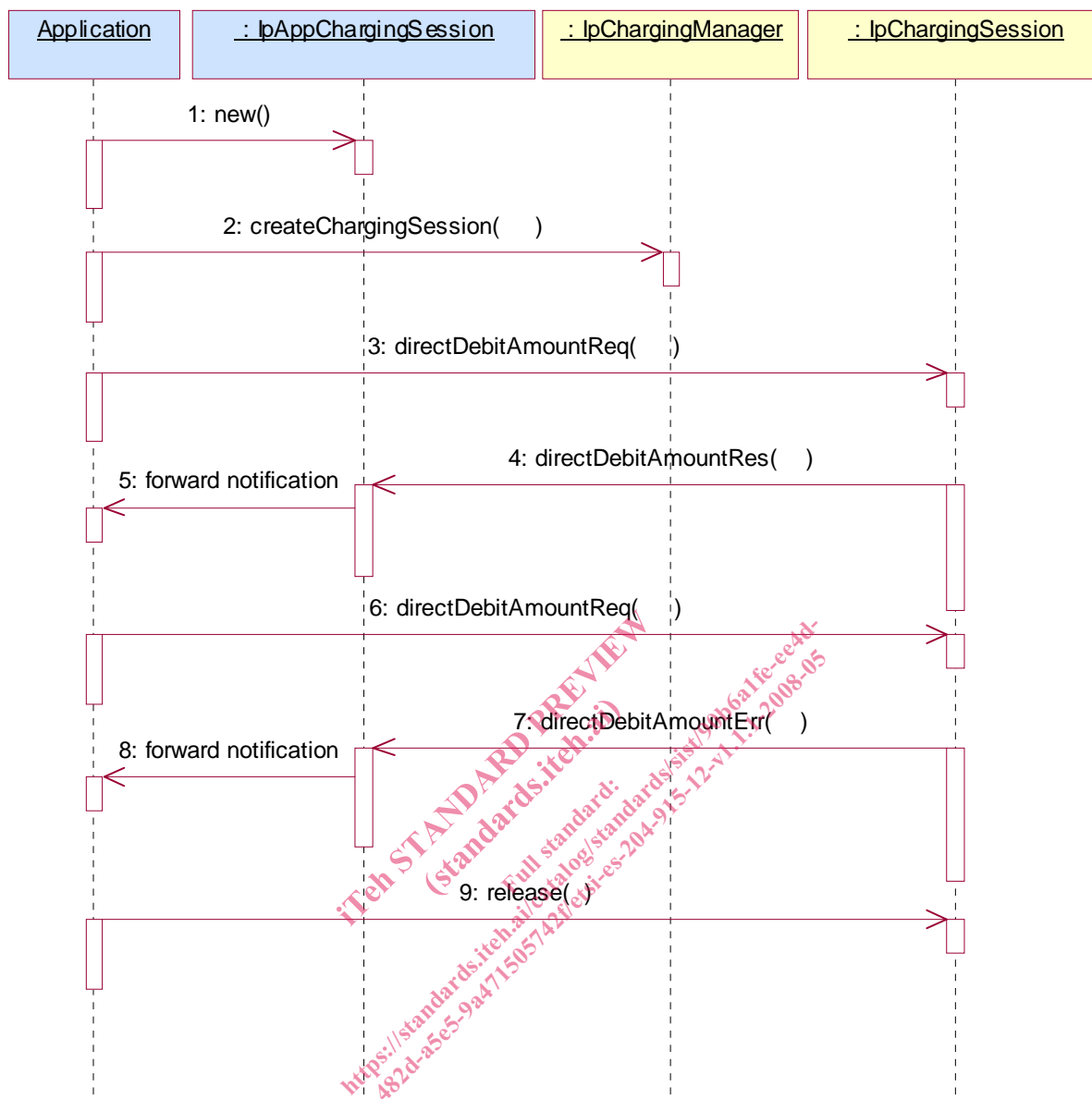
After half of the video has been sent to the user, the application may choose to capture half of the price already:

- 7: The application requests to debit \$1.00 from the reservation.
- 8: The successful debit is reported back to the application.
- 9: The acknowledge is forwarded to the application.
- 10: The application checks if the remaining lifetime of the reservation will cover the remaining 5 minutes of video. Let us assume, it does not.
- 11: The application asks the IpChargingSession object to extend the lifetime of the reservation.
- 12: Assuming that the application provider is allowed to keep reservations open for longer than 10 minutes, the extendLifeTimeReq() will be honoured and confirmed properly.
- 13: The confirmation is forwarded to the application.
- 14: When the complete video has been transmitted to the user without errors, the application charges another \$1.00.
- 15: The IpChargingSession object acknowledges the successful debit at the IpAppChargingSession callback object.
- 16: The IpAppChargingSession object forwards the acknowledge to the application.
- 17: Since the service is complete, the application frees all resources associated with the reservation and session.

5.2 Immediate Charge

This sequence diagram illustrates how immediate charging is used. Assume a WAP gateway that charges the user \$0.01 per requested URL. Since it is acceptable to lose one tick worth \$0.01, no prior reservations are made. The WAP gateway sends an immediate debit for each requested URL, and should a payment have as result failure, the user is disconnected.

The operations which handle units are used exactly the same, except that the amount of application usage is indicated instead of a price.



- 1: The application creates a local object implementing the IpAppChargingSession interface. This object will receive response messages from the IpChargingSession object.
 - 2: The application orders the creation of a session. No new object is created for the charging session handling in this example implementation.
 - 3: The application requests to charge the user \$0.01.
 - 4: The payment is acknowledged.
 - 5: The acknowledgement is forwarded in the application.
 - 6: The application requests to charge the user \$0.01.
 - 7: The payment is reported to fail.
 - 8: The failure report is forwarded in the application.
- (repeat steps 3 to 5 and 6 to 8 as long as you want to in any order you want to).
- 9: The application releases the session.

6 Class Diagrams

This class diagram shows the application interfaces for charging and their relations to the service interfaces.

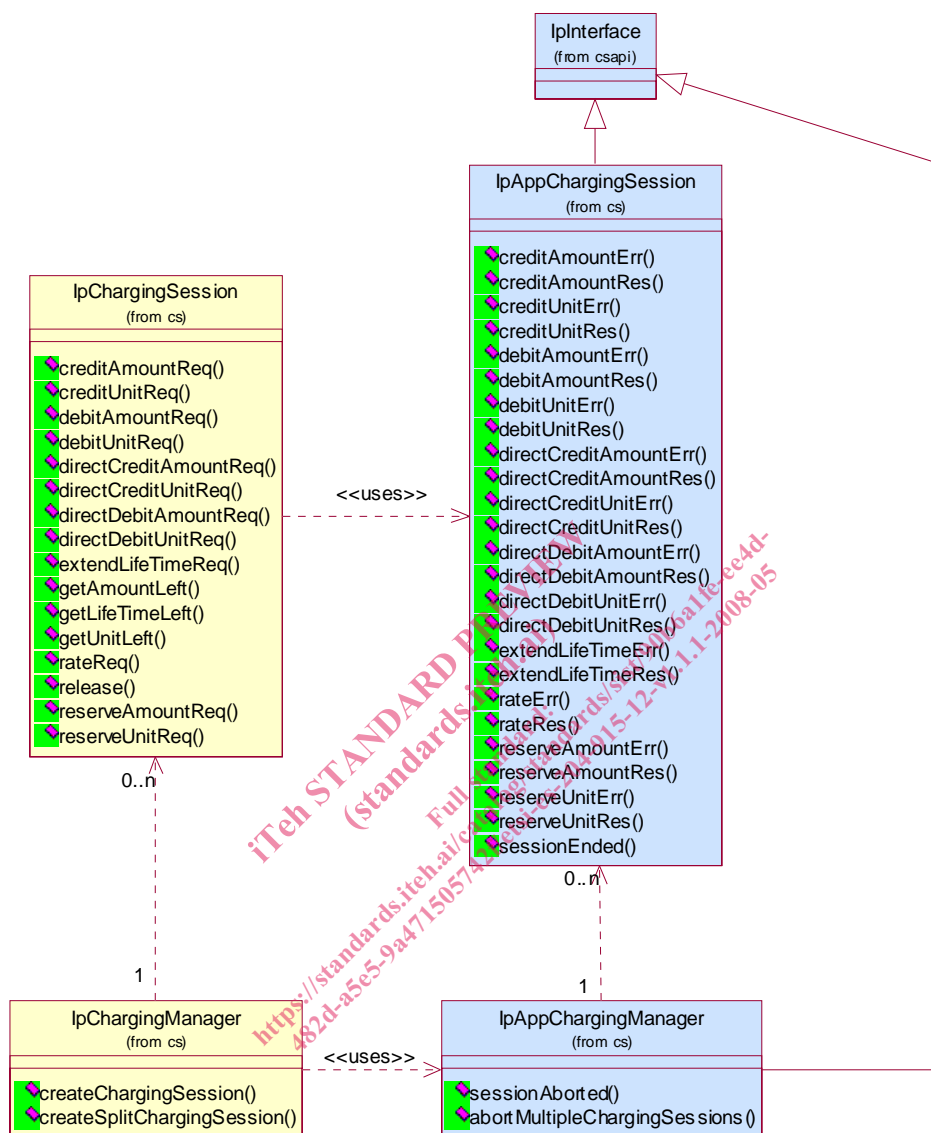


Figure 1: Application Interfaces

This class diagram shows the interfaces of the charging SCF.