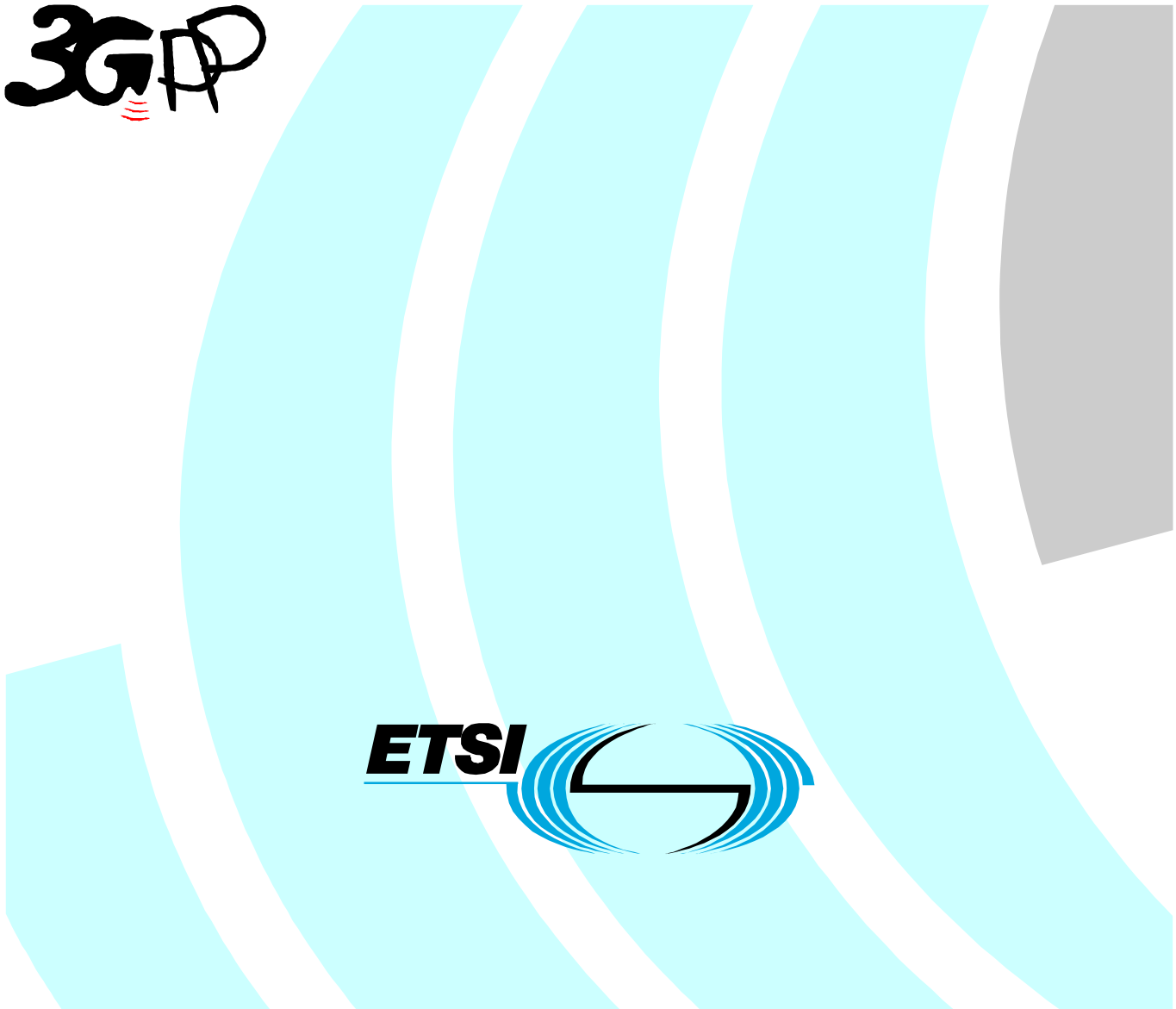
# ETSI TS 129 198-7 V5.0.0 (2002-03)

*Technical Specification*

**Universal Mobile Telecommunications System (UMTS);**
**Open Service Access (OSA);**
**Application Programming Interface (API);**
**Part 7: Terminal capabilities**
**(3GPP TS 29.198-7 version 5.0.0 Release 5)**



Reference
RTS/TSGN-0529198-7Uv5

Keywords
UMTS

***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, send your comment to:
editor@etsi.fr

***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 2002.
All rights reserved.

**DECT**™, **PLUGTESTS**™ and **UMTS**™ are Trade Marks of ETSI registered for the benefit of its Members.
**TIPHON**™ and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
**3GPP**™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

# Intellectual Property Rights

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

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

# Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

# Contents

Intellectual Property Rights ..... 2
Foreword ..... 2
Foreword ..... 5
Introduction ..... 5
1 Scope ..... 7
2 References ..... 7
3 Definitions and abbreviations ..... 8
3.1 Definitions ..... 8
3.2 Abbreviations ..... 8
4 Terminal Capabilities SCF ..... 8
5 Sequence Diagrams ..... 8
5.1 Terminal capabilities example ..... 8
6 Class Diagrams ..... 10
7 The Service Interface Specifications ..... 12
7.1 Interface Specification Format ..... 12
7.1.1 Interface Class ..... 12
7.1.2 Method descriptions ..... 12
7.1.3 Parameter descriptions ..... 12
7.1.4 State Model ..... 12
7.2 Base Interface ..... 13
7.2.1 Interface Class IpInterface ..... 13
7.3 Service Interfaces ..... 13
7.3.1 Overview ..... 13
7.4 Generic Service Interface ..... 13
7.4.1 Interface Class IpService ..... 13
7.4.1.1 Method setCallback() ..... 13
7.4.1.1.1 Parameters ..... 13
7.4.1.1.2 Raises ..... 14
7.4.1.2 Method setCallbackWithSessionID() ..... 14
7.4.1.2.1 Parameters ..... 14
7.4.1.2.2 Raises ..... 14
8 Terminal Capabilities Interface Classes ..... 14
8.1 Interface Class IpTerminalCapabilities ..... 14
8.1.1 Method getTerminalCapabilities() ..... 15
8.1.1.1 Parameters ..... 15
8.1.1.2 Returns ..... 15
8.1.1.3 Raises ..... 15
8.2 Interface Class IpExtendedTerminalCapabilities ..... 15
8.2.1 Method triggeredTerminalCapabilityStartReq() ..... 15
8.2.1.1 Parameters ..... 16
8.2.1.2 Returns ..... 16
8.2.1.3 Raises ..... 16
8.2.2 Method triggeredTerminalCapabilityStop() ..... 16
8.2.2.1 Parameters ..... 16
8.2.2.2 Raises ..... 17
8.3 Interface Class IpAppExtendedTerminalCapabilities ..... 17
8.3.1 Method triggeredTerminalCapabilityReport() ..... 17
8.3.1.1 Parameters ..... 17
8.3.2 Method triggeredTerminalCapabilityReportErr() ..... 17
8.3.2.1 Parameters ..... 18

9 State Transition Diagrams ..... 18

10 Service Properties ..... 18

11 Terminal Capabilities Data Definitions ..... 18
11.1 terminalIdentity ..... 18
11.2 TpTerminalCapabilities ..... 18
11.3 TpTerminalCapabilitiesError ..... 19
11.4 TpTerminalCapabilityChangeCriteria ..... 19
11.5 TpTerminalCapabilityScopeType ..... 19
11.6 TpTerminalCapabilityScope ..... 19

12 Exception Classes ..... 20

**Annex A (normative): OMG IDL Description of Terminal Capabilities SCF ..... 21**

**Annex B (informative): Differences between this draft and 3GPP TS 29.198 Rel-4 ..... 22**

**Annex C (informative): Change history ..... 23**

History ..... 24

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

# Introduction

The present document is part 7 of a multi-part TS covering the 3rd Generation Partnership Project: Technical Specification Group Core Network; Open Service Access (OSA); Application Programming Interface (API), as identified below. The **API specification** (3GPP TS 29.198) is structured in the following Parts:

| | | |
|---|---|---|
| Part 1: | Overview | |
| Part 2: | Common Data Definitions | |
| Part 3: | Framework | |
| Part 4: | Call Control SCF | |
| 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 | (not part of 3GPP Release 5) |
| Part 10: | Connectivity Manager SCF | (not part of 3GPP Release 5) |
| Part 11: | Account Management SCF | |
| Part 12: | Charging SCF | |
| Part 13 : | Policy Management SCF | |
| Part 14 : | Presence and Availability Management SCF | |

The **Mapping specification of the OSA APIs and network protocols** (3GPP TR 29.998) is also structured as above. A mapping to network protocols is however not applicable for all Parts, but the numbering of Parts is kept. Also in case a Part is not supported in a Release, the numbering of the parts is maintained.

| OSA API specifications 29.198-family | | OSA API Mapping - 29.998-family | |
|---|---|---|---|
| **29.198-1** | **Part 1: Overview** | **29.998-1** | **Part 1: Overview** |
| **29.198-2** | **Part 2: Common Data Definitions** | *29.998-2* | *Not Applicable* |
| **29.198-3** | **Part 3: Framework** | *29.998-3* | *Not Applicable* |
| **29.198-4** | **Part 4: Call Control SCF** | **29.998-4-1** | **Subpart 1: Generic Call Control – CAP mapping** |
| | | *29.998-4-2* | |
| **29.198-5** | **Part 5: User Interaction SCF** | **29.998-5-1** | **Subpart 1: User Interaction – CAP mapping** |
| | | *29.998-5-2* | |
| | | *29.998-5-3* | |
| | | **29.998-5-4** | **Subpart 4: User Interaction – SMS mapping** |
| **29.198-6** | **Part 6: Mobility SCF** | **29.998-6** | **User Status and User Location – MAP mapping** |
| **29.198-7** | **Part 7: Terminal Capabilities SCF** | *29.998-7* | *Not Applicable* |
| **29.198-8** | **Part 8: Data Session Control SCF** | **29.998-8** | **Data Session Control – CAP mapping** |
| *29.198-9* | *Part 9: Generic Messaging SCF* | *29.998-9* | *Not Applicable* |
| *29.198-10* | *Part 10: Connectivity Manager SCF* | *29.998-10* | *Not Applicable* |
| **29.198-11** | **Part 11: Account Management SCF** | *29.998-11* | *Not Applicable* |

| | | | |
|---|---|---|---|
| **29.198-12** | **Part 12: Charging SCF** | *29.998-12* | *Not Applicable* |
| **29.198-13** | **Part 13: Policy Management SCF** | *29.998-13* | *Not Applicable* |
| **29.198-14** | **Part 14: Presence and Availability Management SCF** | *29.998-14* | *Not Applicable* |

# 1 Scope

The present document is part 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 concepts and the functional architecture for the OSA are contained in 3GPP TS 23.127 [3]. The requirements for OSA are contained in 3GPP TS 22.127 [2].

The present document specifies the Terminal Capabilities Service Capability Feature (SCF) aspects of the interface. All aspects of the Terminal Capabilities 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

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

This specification has been defined jointly between 3GPP TSG CN WG5, ETSI SPAN 12 and the Parlay Consortium, in co-operation with a number of JAIN™ Community member companies.

# 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. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 29.198-1 "Open Service Access; Application Programming Interface; Part 1: Overview".

[2] 3GPP TS 22.127: "Stage 1 Service Requirement for the Open Service Access (OSA) (Release 5)".

[3] 3GPP TS 23.127: "Virtual Home Environment (Release 5)".

[4] World Wide Web Consortium Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation (www.w3.org).

[5] Wireless Application Protocol (WAP), UAProf Specification (www.wapforum.org).

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 29.198-1 [1] apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 29.198-1 [1] apply.

# 4 Terminal Capabilities SCF

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

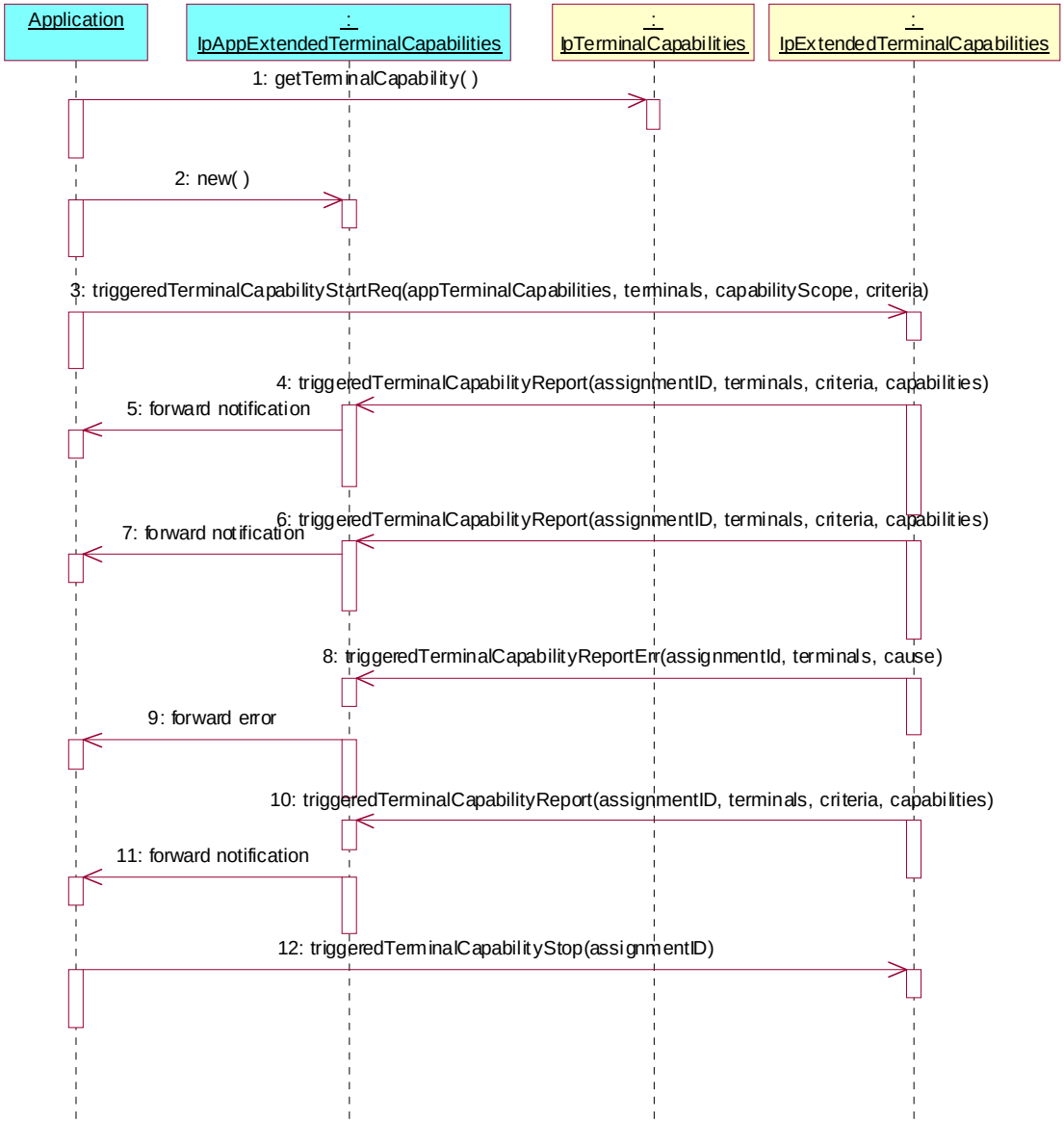
The order is as follows:

- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause show 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 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 section show 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 of this specification.

# 5 Sequence Diagrams

## 5.1 Terminal capabilities example

The following example sequence diagram illustrates how the terminal capabilities can be retrieved and their changes monitored.




1: The application retrieves the terminal capability of a terminal.

2: The application creates an object to implement IpAppExtendedTerminalCapabilities.

3: The terminal capabilities changes are started to be monitored.

4: The terminal capabilities have changed and they are reported as requested.

5: The report is forwarded internally to the application.

6: The terminal capabilities have changed and they are reported as requested.

7: The report is forwarded internally to the application.

8: An error has happened in the monitoring and it is reported.

9: The error report is forwarded internally to the application.

10: The terminal capabilities have changed and they are reported as requested.

11: The report is forwarded internally to the application.

12: The terminal capability monitoring is stopped.

# 6 Class Diagrams

Terminal Capabilities Class Diagram:

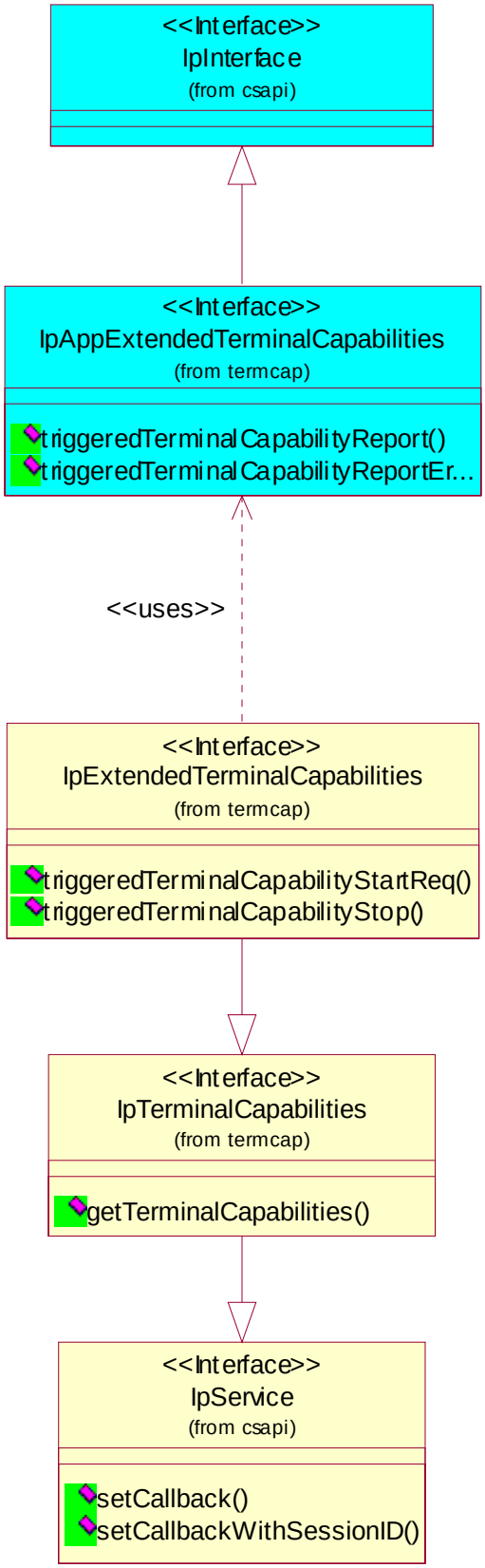

<<Interface>>
IpInterface
(from csapi)
<<Interface>>
IpAppExtendedTerminalCapabilities
(from termcap)
triggeredTerminalCapabilityReport()
triggeredTerminalCapabilityReportEr...
<<uses>>
<<Interface>>
IpExtendedTerminalCapabilities
(from termcap)
triggeredTerminalCapabilityStartReq()
triggeredTerminalCapabilityStop()
<<Interface>>
IpTerminalCapabilities
(from termcap)
getTerminalCapabilities()
<<Interface>>
IpService
(from csapi)
setCallback()
setCallbackWithSessionID()

**Figure: Terminal Capabilities Class Diagram**

# 7 The Service Interface Specifications

## 7.1 Interface Specification Format

This clause defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described below.

### 7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name `Ip<name>`. The callback interfaces to the applications are denoted by classes with name `IpApp<name>`. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name IpSvc<name>, while the Framework interfaces are denoted by classes with name IpFw<name>

### 7.1.2 Method descriptions

Each method (API method “call”) is described. Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a '`Req`' suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a '`Res`' or '`Err`' suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant `IpApp<name> or IpSvc<name>` interfaces to provide the callback mechanism.

### 7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as 'in' represent those that must have a value when the method is called. Those described as 'out' are those that contain the return result of the method when the method returns.

### 7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

# 7.2 Base Interface

## 7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.

| <<Interface>><br>IpInterface |
|---|
| |
| |

# 7.3 Service Interfaces

## 7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

# 7.4 Generic Service Interface

## 7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.

| <<Interface>><br>IpService |
|---|
| |
| setCallback (appInterface : in IpInterfaceRef) : void<br>setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : void |

### 7.4.1.1 Method setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionIDs.

#### 7.4.1.1.1 Parameters

**`appInterface : in IpInterfaceRef`**

Specifies a reference to the application interface, which is used for callbacks

#### 7.4.1.1.2 Raises

**TpCommonExceptions, P_INVALID_INTERFACE_TYPE**

### 7.4.1.2 Method setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not use SessionIDs.

#### 7.4.1.2.1 Parameters

**appInterface : in IpInterfaceRef**

Specifies a reference to the application interface, which is used for callbacks

**sessionID : in TpSessionID**

Specifies the session for which the service can invoke the application's callback interface.

#### 7.4.1.2.2 Raises

**TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_INTERFACE_TYPE**

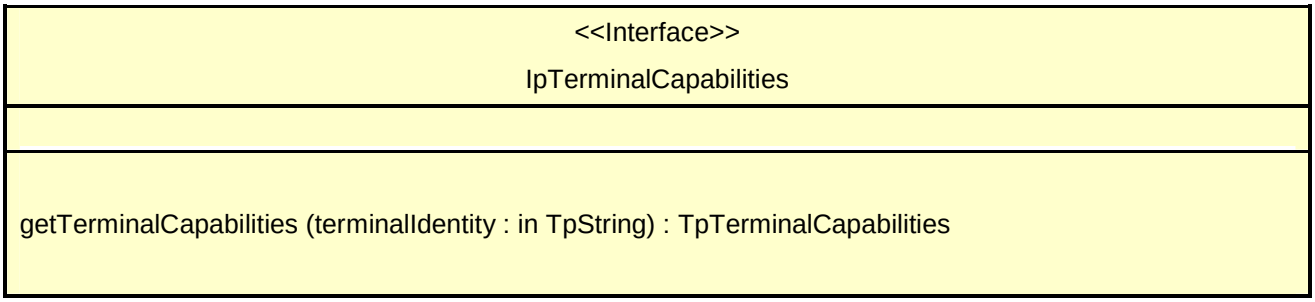
# 8 Terminal Capabilities Interface Classes

The Terminal Capabilities SCF enables the application to retrieve the terminal capabilities of the specified terminal. Additionally it is possible for the application to request notifications when the capabilities of the terminal change in some way. The Terminal Capabilities service provides SCF interfaces IpTerminalCapabilities and IpExtendedTerminalCapabilities. The application side interface for the reporting is called IpAppExtendedTerminalCapabilities.

## 8.1 Interface Class IpTerminalCapabilities

Inherits from: IpService.

The Terminal Capabilities SCF interface IpTerminalCapabilities contains the synchronous method getTerminalCapabilities. The application has to provide the terminaIdentity as input to this method. The result indicates whether or not the terminal capabilities are available in the network and, in case they are, it will return the terminal capabilities (see the data definition of TpTerminalCapabilities for more information). The network may override some capabilities that have been indicated by the terminal itself due to network policies or other restrictions or modifications in the supported capabilities.



### 8.1.1 Method getTerminalCapabilities()

This method is used by an application to get the capabilities of a user's terminal. Direction: Application to Network.

Returns result : Specifies the latest available capabilities of the user´s terminal.

This information, if available, is returned as CC/PP headers as specified in W3C [1] and adopted in the WAP UAProf specification [2]. It contains URLs; terminal attributes and values, in RDF format; or a combination of both.

#### 8.1.1.1 Parameters

**`terminalIdentity : in TpString`**

Identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy.

#### 8.1.1.2 Returns

**`TpTerminalCapabilities`**

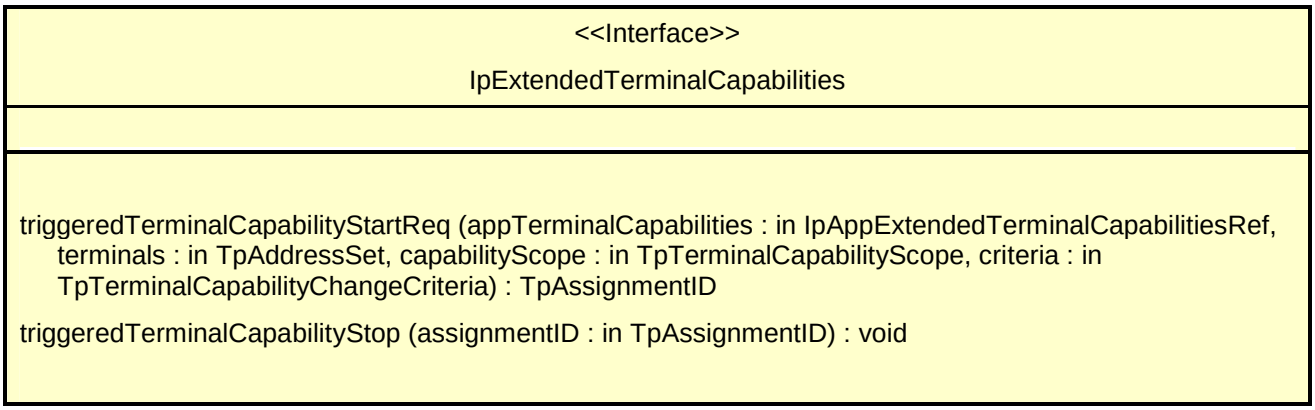
#### 8.1.1.3 Raises

**`TpCommonExceptions, P_INVALID_TERMINAL_ID`**

## 8.2 Interface Class IpExtendedTerminalCapabilities

Inherits from: IpTerminalCapabilities.

This interface can be used as an extended version of terminal capability monitoring. The application programmer can use this interface to request terminal capability reports that are triggered by their changes. Note that the underlying mechanisms for this network feature are currently not fully standardised.




### 8.2.1 Method triggeredTerminalCapabilityStartReq()

Request for terminal capability reports when the capabilities change or when the application obviously does not have the current terminal capability information when this method is invoked.

Returns: assignmentID

Specifies the assignment ID of the triggered terminal capability reporting request.

#### 8.2.1.1 Parameters

**appTerminalCapabilities : in IpAppExtendedTerminalCapabilitiesRef**

Specifies the application interface for callbacks.

**terminals : in TpAddressSet**

Specifies the terminal(s) for which the capabilities shall be reported. TpAddress fields have the following use:

- Plan: Used to indicate the numbering plan
- AddrString: Used to indicate the subscriber address
- Name: Used to indicate the terminal identity. May be applied also together with AddrString to indicate subscriber's particular terminal. The precise format is not defined.
- Presentation: No defined use
- Screening: No defined use
- SubAddressString: No defined use

Hence it is possible to indicate the subscriber and/or the terminal identification. This terminal addressing is implementation specific e.g. subscriber identification may not always be sufficient information to get the capabilities of the terminal.

**capabilityScope : in TpTerminalCapabilityScope**

Specifies the scope of the capabilities that the application is interested in. The contents are implementation specific. One possibility is to use the CC/PP definitions as in TpTerminalCapabilities.

**criteria : in TpTerminalCapabilityChangeCriteria**

Specifies the trigger conditions for the reports e.g. software or hardware update.

#### 8.2.1.2 Returns

**TpAssignmentID**

#### 8.2.1.3 Raises

**TpCommonExceptions, P_INFORMATION_NOT_AVAILABLE, P_INVALID_INTERFACE_TYPE, P_INVALID_CRITERIA, P_INVALID_TERMINAL_ID**

### 8.2.2 Method triggeredTerminalCapabilityStop()

Stop reporting for terminal capability changes that were started by triggeredTerminalCapabilityStartReq().

#### 8.2.2.1 Parameters

**assignmentID : in TpAssignmentID**

Specifies the assignment ID for the task to be stopped.

#### 8.2.2.2 Raises

**TpCommonExceptions, P_INVALID_ASSIGNMENT_ID**

## 8.3 Interface Class IpAppExtendedTerminalCapabilities

Inherits from: IpInterface.

IpAppExtendedTerminalCapabilities interface is used to send triggered terminal capability reports. It is implemented by the client application developer.

| <<Interface>><br>IpAppExtendedTerminalCapabilities |
|---|
| |
| triggeredTerminalCapabilityReport (assignmentID : in TpAssignmentID, terminals : in TpAddressSet, criteria : in TpTerminalCapabilityChangeCriteria, capabilities : in TpTerminalCapabilities) : void<br>triggeredTerminalCapabilityReportErr (assignmentId : in TpAssignmentID, terminals : in TpAddressSet, cause : in TpTerminalCapabilitiesError) : void |

### 8.3.1 Method triggeredTerminalCapabilityReport()

This terminal capability report is issued when the capabilities of the terminal have changed in the way specified by the criteria parameter in the previously invoked triggeredTerminalCapabilityStartReq () method.

#### 8.3.1.1 Parameters

**assignmentID : in TpAssignmentID**

Specifies the assignment ID of the report.

**terminals : in TpAddressSet**

Specifies the terminal(s) either by subscriber or terminal ID or both as described for the triggeredTerminalCapabilityStartReq () method.

**criteria : in TpTerminalCapabilityChangeCriteria**

Specifies the criteria that caused the report to be sent.

**capabilities : in TpTerminalCapabilities**

Specifies the capabilities of the terminal. The network may override some capabilities that have been indicated by the terminal itself due to network policies or other restrictions or modifications in the supported capabilities.

### 8.3.2 Method triggeredTerminalCapabilityReportErr()

This method indicates that the requested reporting has failed. Note that errors may concern the whole assignment or just some terminals. In the former case no terminals are specified.

#### 8.3.2.1 Parameters

**`assignmentId : in TpAssignmentID`**

Specifies the assignment ID.

**`terminals : in TpAddressSet`**

Specifies the terminal(s) either by subscriber or terminal ID or both as described for the triggeredTerminalCapabilityStartReq () method.

**`cause : in TpTerminalCapabilitiesError`**

Specifies the error that led to the failure.

# 9 State Transition Diagrams

There are no State Transition Diagrams for the Terminal Capabilities SCF.

# 10 Service Properties

The following table lists properties relevant for this SCF.

| Property | Type | Description |
|---|---|---|
| P_TRIGGERED_REPORTING_SUPPORTED | BOOLEAN_SET | Value = TRUE : The triggered reporting of terminal capabilities is supported by the SCF.<br>Value = FALSE : The triggered reporting of terminal capabilities is not supported by the SCF. |

# 11 Terminal Capabilities Data Definitions

All data types referenced but not defined in this clause are common data definitions which may be found in 3GPP TS 29.198-2.

## 11.1 terminalIdentity

Identifies the terminal.

| Name | Type | Documentation |
|---|---|---|
| `terminalIdentity` | `TpString` | Identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy. |

## 11.2 TpTerminalCapabilities

This data type is a Sequence of Data Elements that describes the terminal capabilities. It is a structured type that consists of:

| Sequence Element Name | Sequence Element Type | Documentation |
|---|---|---|
| `StatusCode` | `TpBoolean` | Indicates whether or not the TerminalCapabilities are available. |
| `TerminalCapabilities` | `TpString` | Specifies the latest available capabilities of the user's terminal.<br>This information, if available, is returned as CC/PP headers as specified in W3C (see [6] in ES 201 915-1)and adopted in the WAP UAProf specification (see [9] in ES 201 915-1). It contains URLs; terminal attributes and values, in RDF format; or a combination of both. |

## 11.3 TpTerminalCapabilitiesError

Defines an error that is reported by the Terminal Capabilities SCF.

| Name | Value | Description |
|---|---|---|
| P_TERMCAP_ERROR_UNDEFINED | 0 | Undefined. |
| P_TERMCAP_INVALID_TERMINALID | 1 | The request can not be handled because the terminal id specified is not valid. |
| P_TERMCAP_SYSTEM_FAILURE | 2 | System failure.<br>The request cannot be handled because of a general problem in the terminal capabilities service or the underlying network. |
| P_TERMCAP_INFO_UNAVAILABLE | 3 | The terminal capability information is not available. |

## 11.4 TpTerminalCapabilityChangeCriteria

Defines the type of the terminal capability changes to be reported. The values may be combined by a logical 'OR' function.

| Name | Value | Description |
|---|---|---|
| P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_UNDEFINED | 00h | Undefined |
| P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_GENERAL | 01h | Any change in the terminal capabilities. |
| P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_HW_UPDATE | 02h | The terminal device hardware has been modified or replaced completely. |
| P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_SW_UPDATE | 04h | The software of the terminal has been updated in any way. Also changes in configuration or preferences may be concerned. |
| P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_INITIAL | 08h | The initial device capabilities reported when monitoring has been started by an application. |

## 11.5 TpTerminalCapabilityScopeType

Defines a specific type of the terminal capability scope definition.

| Name | Value | Description |
|---|---|---|
| P_TERMINAL_CAPABILITY_SCOPE_TYPE_UNDEFINED | 0 | Undefined. |
| P_TERMINAL_CAPABILITY_SCOPE_TYPE_CCPP | 1 | Indicates that the terminal capability scope is expressed as CC/PP headers as specified in W3C [4] and adopted in the WAP UAProf specification [5]. It contains URLs; terminal attributes and values, in RDF format; or a combination of both. |

## 11.6 TpTerminalCapabilityScope

Defines the `Sequence of Data Elements` that specify the scope of the terminal capabilities.

| Sequence Element Name | Sequence Element Type |
|---|---|
| ScopeType | TpTerminalCapabilityScopeType |
| Scope | TpString |

# 12 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

| Name | Description |
|---|---|
| P_INVALID_TERMINAL_ID | The request can not be handled because the terminal id specified is not valid. |

Each exception class contains the following structure:

| Structure Element Name | Structure Element Type | Structure Element Description |
|---|---|---|
| ExtraInformation | TpString | Carries extra information to help identify the source of the exception, e.g. a parameter name |

# Annex A (normative): OMG IDL Description of Terminal Capabilities SCF

The OMG IDL representation of this interface specification is contained in a text file (termcap.idl contained in archive 2919807IDL.ZIP) which accompanies the present document.

# Annex B (informative): Differences between this draft and 3GPP TS 29.198 Rel-4

New interfaces IpExtendedTerminalCapabilities, IpAppExtendedTerminalCapabilities added.

# Annex C (informative): Change history

| Change history | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Date** | **TSG #** | **TSG Doc.** | **CR** | **Rev** | **Subject/Comment** | **Old** | **New** |
| Mar 2001 | CN_11 | NP-010134 | 047 | -- | CR 29.198: for moving TS 29.198 from R99 to Rel 4 (N5-010158) | 3.2.0 | 4.0.0 |
| Jun 2001 | CN_12 | NP-010330 | 001 | -- | Corrections to OSA API Rel4 | 4.0.0 | 4.1.0 |
| Sep 2001 | CN_13 | NP-010470 | 002 | -- | Changing references to JAIN | 4.1.0 | 4.2.0 |
| Dec 2001 | CN_14 | NP-010600 | 003 | -- | Replace Out Parameters with Return Types | 4.2.0 | 4.3.0 |
| Mar 2002 | CN_15 | NP-020109 | 004 | -- | Add P_INVALID_INTERFACE_TYPE exception to IpService.setCallback() and IpService.setCallbackWithSessionID() | 4.3.0 | 4.4.0 |
| Mar 2002 | CN_15 | NP-020113 | 005 | -- | Addition of terminal capability change notifications | 4.4.0 | 5.0.0 |
| | | | | | | | |

# History

| Document history | | |
|---|---|---|
| V5.0.0 | March 2002 | Publication |
| | | |
| | | |
| | | |
| | | |