



SLOVENSKI STANDARD
SIST ES 201 915-3 V1.4.1:2005
01-januar-2005

Odpri dostop do storitve (OSA) – Vmesnik za aplikacijsko programiranje (API) – 3.
del: Okvir

Open Service Access (OSA); Application Programming Interface (API); Part 3:
Framework (Parlay 3)

iTeh STANDARD PREVIEW
(standards.iteh.ai)

Ta slovenski standard je istoveten z: **ES 201 915-3 Version 1.4.1**
<https://standards.iteh.ai/catalog/standards/sist/f7e1bc9c-3d15-4910-825c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005>

ICS:

33.040.01	Telekomunikacijski sistemi na splošno	Telecommunication systems in general
-----------	--	---

SIST ES 201 915-3 V1.4.1:2005 **en**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST ES 201 915-3 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005>

ETSI ES 201 915-3 V1.4.1 (2003-07)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework (Parlay 3)



iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST ES 201 915-3 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005>



Reference

RES/SPAN-120095-3

Keywords

API, OSA, IDL, 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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST ES 201 915-3 V1.4.1:2005

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287d000/sist-201-915-3-v1-4-1-2005>
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.org

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 2003.

© The Parlay Group 2003.

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.

Contents

Intellectual Property Rights	9
Foreword.....	9
1 Scope	10
2 References	10
3 Definitions and abbreviations.....	10
3.1 Definitions	10
3.2 Abbreviations	10
4 Overview of the Framework.....	10
4.1 General requirements on support of methods.....	12
5 The Base Interface Specification.....	12
5.1 Interface Specification Format	12
5.1.1 Interface Class	12
5.1.2 Method descriptions.....	13
5.1.3 Parameter descriptions.....	13
5.1.4 State Model.....	13
5.2 Base Interface.....	13
5.2.1 Interface Class IpInterface	13
5.3 Service Interfaces	13
5.3.1 Overview	13
5.4 Generic Service Interface	13
5.4.1 Interface Class IpService	13
6 Framework Access Session API.....	15
6.1 Sequence Diagrams	15
6.1.1 Trust and Security Management Sequence Diagrams	15
6.1.1.1 Initial Access for trusted parties.....	15
6.1.1.2 Initial Access.....	15
6.1.1.3 Authentication.....	17
6.1.1.4 API Level Authentication	17
6.2 Class Diagrams.....	19
6.3 Interface Classes.....	19
6.3.1 Trust and Security Management Interface Classes	19
6.3.1.1 Interface Class IpClientAPILevelAuthentication.....	20
6.3.1.2 Interface Class IpClientAccess.....	21
6.3.1.3 Interface Class IpInitial.....	22
6.3.1.4 Interface Class IpAuthentication.....	23
6.3.1.5 Interface Class IpAPILevelAuthentication	24
6.3.1.6 Interface Class IpAccess	26
6.4 State Transition Diagrams	28
6.4.1 Trust and Security Management State Transition Diagrams	29
6.4.1.1 State Transition Diagrams for IpInitial	29
6.4.1.1.1 Active State	29
6.4.1.1.2 State Transition Diagrams for IpAPILevelAuthentication.....	29
6.4.1.1.2.1 Idle State.....	30
6.4.1.1.2.2 Selecting Method State.....	30
6.4.1.1.2.3 Authenticating Client State.....	30
6.4.1.1.2.4 Client Authenticated State	30
6.4.1.1.3 State Transition Diagrams for IpAccess.....	31
6.4.1.1.3.1 Active State	31
6.4.1.3.1 Active State	31
7 Framework-to-Application API	32
7.1 Sequence Diagrams	32
7.1.1 Event Notification Sequence Diagrams	32
7.1.1.1 Enable Event Notification	32

7.1.2	Integrity Management Sequence Diagrams	33
7.1.2.1	Load Management: Suspend/resume notification from application	33
7.1.2.2	Load Management: Framework queries load statistics	34
7.1.2.3	Load Management: Framework callback registration and Application load control	35
7.1.2.4	Load Management: Application reports current load condition	36
7.1.2.5	Load Management: Application queries load statistics	36
7.1.2.6	Load Management: Application callback registration and load control	37
7.1.2.7	Heartbeat Management: Start/perform/end heartbeat supervision of the application	38
7.1.2.8	Fault Management: Framework detects a Service failure	39
7.1.2.9	Fault Management: Application requests a Framework activity test	40
7.1.3	Service Discovery Sequence Diagrams	40
7.1.3.1	Service Discovery	40
7.1.4	Service Agreement Management Sequence Diagrams	42
7.1.4.1	Service Selection	42
7.2	Class Diagrams	44
7.3	Interface Classes	47
7.3.1	Service Discovery Interface Classes	47
7.3.1.1	Interface Class IpServiceDiscovery	47
7.3.2	Service Agreement Management Interface Classes	50
7.3.2.1	Interface Class IpAppServiceAgreementManagement	50
7.3.2.2	Interface Class IpServiceAgreementManagement	52
7.3.3	Integrity Management Interface Classes	55
7.3.3.1	Interface Class IpAppFaultManager	55
7.3.3.2	Interface Class IpFaultManager	58
7.3.3.3	Interface Class IpAppHeartBeatMgmt	61
7.3.3.4	Interface Class IpAppHeartBeat	62
7.3.3.5	Interface Class IpHeartBeatMgmt	63
7.3.3.6	Interface Class IpHeartBeat	64
7.3.3.7	Interface Class IpAppLoadManager	65
7.3.3.8	Interface Class IpLoadManager	67
7.3.3.9	Interface Class IpOAM	71
7.3.3.10	Interface Class IpAppOAM	72
7.3.4	Event Notification Interface Classes	72
7.3.4.1	Interface Class IpAppEventNotification	72
7.3.4.2	Interface Class IpEventNotification	73
7.4	State Transition Diagrams	74
7.4.1	Service Discovery State Transition Diagrams	75
7.4.1.1	State Transition Diagrams for IpServiceDiscovery	75
7.4.1.1.1	Active State	75
7.4.2	Service Agreement Management State Transition Diagrams	75
7.4.3	Integrity Management State Transition Diagrams	76
7.4.3.1	State Transition Diagrams for IpLoadManager	76
7.4.3.1.1	Idle State	76
7.4.3.1.2	Notification Suspended State	76
7.4.3.1.3	Active State	76
7.4.3.2	State Transition Diagrams for LoadManagerInternal	77
7.4.3.2.1	Normal load State	77
7.4.3.2.2	Application Overload State	77
7.4.3.2.3	Internal overload State	77
7.4.3.2.4	Internal and Application Overload State	77
7.4.3.3	State Transition Diagrams for IpOAM	78
7.4.3.3.1	Active State	78
7.4.3.4	State Transition Diagrams for IpFaultManager	78
7.4.3.4.1	Framework Active State	79
7.4.3.4.2	Framework Faulty State	79
7.4.3.4.3	Framework Activity Test State	79
7.4.3.4.4	Service Activity Test State	79
7.4.4	Event Notification State Transition Diagrams	79
7.4.4.1	State Transition Diagrams for IpEventNotification	79
7.4.4.1.1	Idle State	80
7.4.4.1.2	Notification Active State	80

8	Framework-to-Enterprise Operator API.....	80
8.1	Sequence Diagrams	84
8.1.1	Service Subscription Sequence Diagrams.....	84
8.1.1.1	Service Discovery and Subscription Scenario.....	84
8.1.1.2	Enterprise Operator and Client Application Subscription Management Sequence Diagram	85
8.2	Class Diagrams.....	88
8.3	Interface Classes.....	89
8.3.1	Service Subscription Interface Classes	89
8.3.1.1	Interface Class IpClientAppManagement	89
8.3.1.2	Interface Class IpClientAppInfoQuery	93
8.3.1.3	Interface Class IpServiceProfileManagement	95
8.3.1.4	Interface Class IpServiceProfileInfoQuery	98
8.3.1.5	Interface Class IpServiceContractManagement	99
8.3.1.6	Interface Class IpServiceContractInfoQuery	101
8.3.1.7	Interface Class IpEntOpAccountManagement	103
8.3.1.8	Interface Class IpEntOpAccountInfoQuery	104
8.4	State Transition Diagrams	104
8.4.1	Service Subscription State Transition Diagrams.....	104
9	Framework-to-Service API	105
9.1	Sequence Diagrams	105
9.1.1	Service Discovery Sequence Diagrams	105
9.1.2	Service Registration Sequence Diagrams	105
9.1.2.1	New SCF Registration.....	105
9.1.3	Service Instance Lifecycle Manager Sequence Diagrams	107
9.1.3.1	Sign Service Agreement.....	107
9.1.4	Integrity Management Sequence Diagrams	108
9.1.4.1	Load Management: Service callback registration and load control.....	108
9.1.4.2	Load Management: Framework callback registration and service load control	109
9.1.4.3	Load Management: Client and Service Load Balancing	110
9.1.4.4	Heartbeat Management: Start/perform/end heartbeat supervision of the service	111
9.1.4.5	Fault Management: Service requests Framework activity test	111
9.1.4.6	Fault Management: Service requests Application activity test	112
9.1.4.7	Fault Management: Application requests Service activity test	113
9.1.4.8	Fault Management: Application detects service is unavailable	114
9.1.5	Event Notification Sequence Diagrams	114
9.2	Class Diagrams.....	115
9.3	Interface Classes.....	117
9.3.1	Service Registration Interface Classes	117
9.3.1.1	Interface Class IpFwServiceRegistration	118
9.3.2	Service Instance Lifecycle Manager Interface Classes	121
9.3.2.1	Interface Class IpServiceInstanceLifecycleManager	121
9.3.3	Service Discovery Interface Classes	122
9.3.3.1	Interface Class IpFwServiceDiscovery	123
9.3.4	Integrity Management Interface Classes.....	125
9.3.4.1	Interface Class IpFwFaultManager	125
9.3.4.2	Interface Class IpSvcFaultManager	128
9.3.4.3	Interface Class IpFwHeartBeatMgmt.....	133
9.3.4.4	Interface Class IpFwHeartBeat	134
9.3.4.5	Interface Class IpSvcHeartBeatMgmt.....	134
9.3.4.6	Interface Class IpSvcHeartBeat	136
9.3.4.7	Interface Class IpFwLoadManager	136
9.3.4.8	Interface Class IpSvcLoadManager	140
9.3.4.9	Interface Class IpFwOAM	142
9.3.4.10	Interface Class IpSvcOAM	143
9.3.5	Event Notification Interface Classes.....	144
9.3.5.1	Interface Class IpFwEventNotification	144
9.3.5.2	Interface Class IpSvcEventNotification	145
9.4	State Transition Diagrams	146
9.4.1	Service Registration State Transition Diagrams	147
9.4.1.1	State Transition Diagrams for IpFwServiceRegistration.....	147
9.4.1.1.1	SCF Registered State	147

9.4.1.1.2	SCF Announced State.....	147
9.4.2	Service Instance Lifecycle Manager State Transition Diagrams	147
9.4.3	Service Discovery State Transition Diagrams	148
9.4.4	Integrity Management State Transition Diagrams	148
9.4.4.1	State Transition Diagrams for IpFwLoadManager.....	148
9.4.4.1.1	Idle State.....	148
9.4.4.1.2	Notification Suspended State.....	148
9.4.4.1.3	Active State	148
9.4.5	Event Notification State Transition Diagrams	149
10	Service Properties.....	149
10.1	Service Property Types	149
10.2	General Service Properties	150
10.2.1	Service Name.....	150
10.2.2	Service Version.....	150
10.2.3	Service Instance ID.....	150
10.2.4	Service Instance Description.....	150
10.2.5	Product Name	150
10.2.6	Product Version	150
10.2.7	Supported Interfaces	150
10.2.8	Operation Set	150
11	Data Definitions	151
11.1	Common Framework Data Definitions	151
11.1.1	TpClientAppID.....	151
11.1.2	TpClientAppIDList.....	151
11.1.3	TpDomainID.....	151
11.1.4	TpDomainIDType.....	152
11.1.5	TpEntOpID	152
11.1.6	TpPropertyName.....	152
11.1.7	TpPropertyValue.....	152
11.1.8	TpProperty	152
11.1.9	TpPropertyList.....	152
11.1.10	TpEntOpIDList.....	152
11.1.11	TpFwID	152
11.1.12	TpService.....	153
11.1.13	TpServiceList.....	153
11.1.14	TpServiceDescription	153
11.1.15	TpServiceID.....	153
11.1.16	TpServiceIDList	153
11.1.17	TpServiceInstanceID	153
11.1.18	TpServiceSpecString	153
11.1.19	TpServiceTypeProperty	154
11.1.20	TpServiceTypePropertyList.....	154
11.1.21	TpServiceTypePropertyMode.....	154
11.1.22	TpServicePropertyTypeName.....	154
11.1.23	TpServicePropertyName.....	154
11.1.24	TpServicePropertyNameList.....	154
11.1.25	TpServicePropertyValue	154
11.1.26	TpServicePropertyValueList.....	154
11.1.27	TpServiceProperty	155
11.1.28	TpServicePropertyList	155
11.1.29	TpServiceSupplierID	155
11.1.30	TpServiceTypeDescription	155
11.1.31	TpServiceTypeName	155
11.1.32	TpServiceTypeNameList	156
11.1.33	TpSubjectType.....	156
11.2	Event Notification Data Definitions	156
11.2.1	TpFwEventName	156
11.2.2	TpFwEventCriteria	156
11.2.3	TpFwEventInfo.....	157
11.3	Trust and Security Management Data Definitions	157

STANDARD PREVIEW
(standards.itech.ai)

SIST ES 201 915-3 V1.4.1:2005

<https://standards.itech.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005>

11.3.1	TpAccessType	157
11.3.2	TpAuthType.....	157
11.3.3	TpEncryptionCapability.....	158
11.3.4	TpEncryptionCapabilityList	158
11.3.5	TpEndAccessProperties	158
11.3.6	TpAuthDomain	158
11.3.7	TpInterfaceName	159
11.3.8	TpInterfaceNameList.....	159
11.3.9	TpServiceToken.....	159
11.3.10	TpSignatureAndServiceMgr	159
11.3.11	TpSigningAlgorithm.....	160
11.4	Integrity Management Data Definitions	160
11.4.1	TpActivityTestRes	160
11.4.2	TpFaultStatsRecord	160
11.4.3	TpFaultStats	160
11.4.4	TpFaultStatisticsError.....	160
11.4.5	TpFaultStatsSet	161
11.4.6	TpActivityTestID.....	161
11.4.7	TpInterfaceFault	161
11.4.8	TpSvcUnavailReason.....	161
11.4.9	TpFwUnavailReason	161
11.4.10	TpLoadLevel.....	161
11.4.11	TpLoadThreshold	162
11.4.12	TpLoadInitVal	162
11.4.13	TpLoadPolicy	162
11.4.14	TpLoadStatistic.....	162
11.4.15	TpLoadStatisticList.....	162
11.4.16	TpLoadStatisticData	162
11.4.17	TpLoadStatisticEntityID.....	163
11.4.18	TpLoadStatisticEntityType.....	163
11.4.19	TpLoadStatisticInfo	163
11.4.20	TpLoadStatisticInfoType	163
11.4.21	TpLoadStatisticError.....	163
11.5	Service Subscription Data Definitions	164
11.5.1	TpPropertyName.....	164
11.5.2	TpPropertyValue.....	164
11.5.3	TpProperty	164
11.5.4	TpPropertyList.....	164
11.5.5	TpEntOpProperties	164
11.5.6	TpEntOp	164
11.5.7	TpServiceContractID.....	164
11.5.8	TpServiceContractIDList.....	164
11.5.9	TpPersonName	164
11.5.10	TpPostalAddress	165
11.5.11	TpTelephoneNumber	165
11.5.12	TpEmail	165
11.5.13	TpHomePage	165
11.5.14	TpPersonProperties	165
11.5.15	TpPerson.....	165
11.5.16	TpServiceStartDate.....	165
11.5.17	TpServiceEndDate.....	165
11.5.18	TpServiceRequestor.....	165
11.5.19	TpBillingContact	165
11.5.20	TpServiceSubscriptionProperties.....	166
11.5.21	TpServiceContract	166
11.5.22	TpServiceContractDescription.....	166
11.5.23	TpClientAppProperties	166
11.5.24	TpClientAppDescription.....	166
11.5.25	TpSagID.....	166
11.5.26	TpSagIDList	167
11.5.27	TpSagDescription	167
11.5.28	TpSag.....	167

STANDARD PREVIEW
(standards.iteh.ai)

SIST ES 201 915-3 V1.4.1:2005

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-11875abb/sist-es-201-915-3-v1-4-1-2005>

11.5.29	TpServiceProfileID	167
11.5.30	TpServiceProfileIDList	167
11.5.31	TpServiceProfile	167
11.5.32	TpServiceProfileDescription	167
12	Exception Classes	168
Annex A (normative):	OMG IDL Description of Framework	169
Annex B (informative):	Contents of 3GPP OSA R4 Framework	170
Annex C (informative):	Record of changes	171
C.1	Interfaces	171
C.1.1	New	171
C.1.2	Deprecated	171
C.1.3	Removed	171
C.2	Methods	172
C.2.1	New	172
C.2.2	Deprecated	172
C.2.3	Modified	172
C.2.4	Removed	172
C.3	Data Definitions	173
C.3.1	New	173
C.3.2	Modified	173
C.3.3	Removed	173
C.4	Service Properties	173
C.4.1	New	173
C.4.2	Deprecated	173
C.4.3	Modified	174
C.4.4	Removed	174
C.5	Exceptions	174
C.5.1	New	174
C.5.2	Modified	174
C.5.3	Removed	174
C.6	Others	174
History	175

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST ES 201 915-3 V1.4.1:2005](https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005)

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005>

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 Services and Protocols for Advanced Networks (SPAN).

The present document is part 3 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 201 915) 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";
- Part 10: "Connectivity Manager SCF";
- Part 11: "Account Management SCF";
- Part 12: "Charging 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 3.3 set of specifications.

A subset of the present document is in 3GPP TS 29.198-3 4.7.0 (Release 4).

1 Scope

The present document is part 3 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 Framework aspects of the interface. All aspects of the Framework are defined in the present document, 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).

iTeh STANDARD PREVIEW

2 References (standards.iteh.ai)

The references listed in clause 2 of ES 201 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

<https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c392a701a01/sist/f7efbc9c-3d13-49f0-823c-6c392a701a01-2005>

ETSI ES 201 915-1: "Open Service Access; Application Programming Interface; Part 1: Overview (Parlay 3)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 201 915-1 apply.

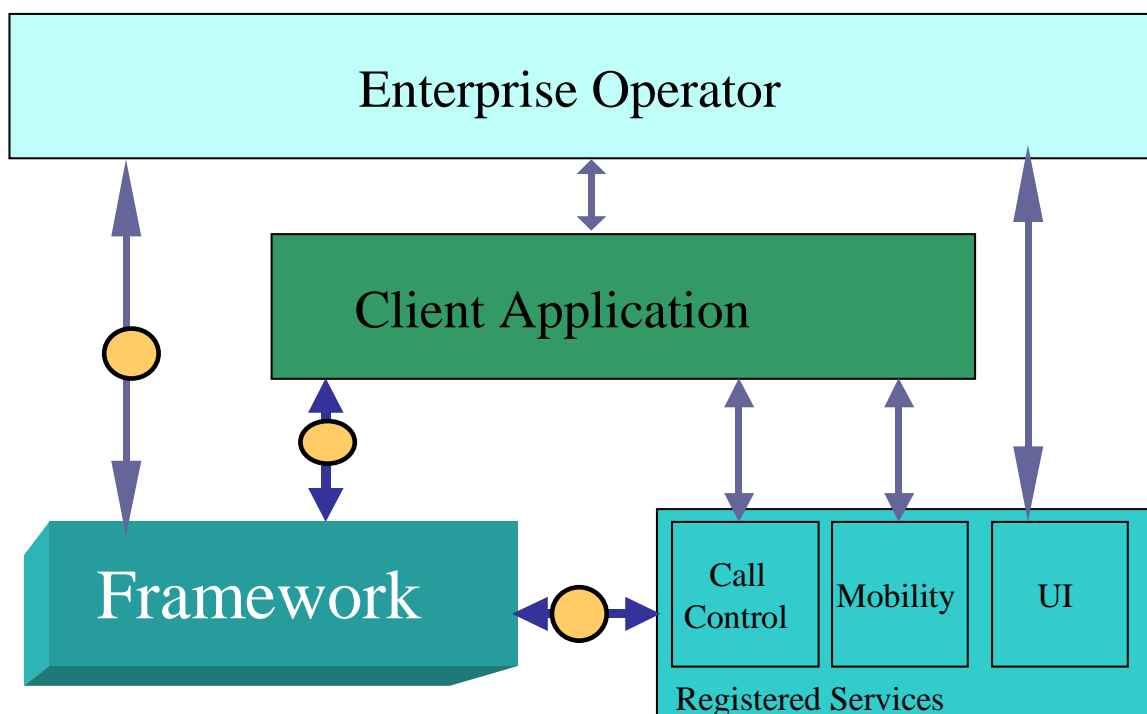
3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ES 201 915-1 apply.

4 Overview of the Framework

This clause explains which basic mechanisms are executed in the OSA Framework prior to offering and activating applications.

The Framework API contains interfaces between the Application Server and the Framework, between the Network Service Capability Server (SCS) and the Framework, and between the Enterprise Operator and the Framework (these interfaces are represented by the yellow circles in the diagram below). The description of the Framework in the present document separates the interfaces into these three distinct sets: Framework to Application interfaces, Framework to Enterprise Operator interfaces and Framework to Service interfaces.



Some of the mechanisms are applied only once (e.g. establishment of service agreement), others are applied each time a user subscription is made to an application (e.g. enabling the call attempt event for a new user).

Basic mechanisms between Application and Framework:

- **Authentication:** Once an off-line service agreement exists, the application can access the authentication interface. The authentication model of OSA is a peer-to-peer model, but authentication does not have to be mutual. The application must be authenticated before it is allowed to use any other OSA interface. It is a policy decision for the application whether it must authenticate the framework or not. It is a policy decision for the framework whether it allows an application to authenticate it before it has completed its authentication of the application.
- **Authorisation:** Authorisation is distinguished from authentication in that authorisation is the action of determining what a previously authenticated application is allowed to do. Authentication must precede authorisation. Once authenticated, an application is authorised to access certain service capability features.
- **Discovery of framework and network service capability features:** After successful authentication, applications can obtain available framework interfaces and use the discovery interface to obtain information on authorised network service capability features. The Discovery interface can be used at any time after successful authentication.
- **Establishment of service agreement:** Before any application can interact with a network service capability feature, a service agreement must be established. A service agreement may consist of an off-line (e.g. by physically exchanging documents) and an on-line part. The application has to sign the on-line part of the service agreement before it is allowed to access any network service capability feature.
- **Access to network service capability features:** The framework must provide access control functions to authorise the access to service capability features or service data for any API method from an application, with the specified security level, context, domain, etc.

Basic mechanism between Framework and Service Capability Server:

- **Registering of network service capability features:** SCFs offered by a Service Capability Server can be registered at the Framework. In this way the Framework can inform the Applications upon request about available service capability features (Discovery). For example, this mechanism is applied when installing or upgrading a Service Capability Server.

Basic mechanism between Framework and Enterprise Operator:

- **Service Subscription function:** This function represents a contractual agreement between the Enterprise Operator and the Framework. In this subscription business model, the enterprise operators act in the role of *subscriber/customer* of services and the client applications act in the role of *users or consumers* of services. The framework itself acts in the role of *retailer* of services.

The following clauses describe each aspect of the Framework in the following order:

- The *sequence diagrams* give the reader a practical idea of how the Framework is implemented.
- The *class diagrams* clause show how each of the interfaces applicable to the Framework 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 Framework. 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 of the present document.

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 Framework or 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 The Base Interface Specification

5.1 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.

5.1.1 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 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>

5.1.2 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.

5.1.3 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.

5.1.4 State Model

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

5.2 Base Interface

5.2.1 Interface Class IpInterface

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

<p>ITeH STANDARD PREVIEW (standards.iteh.ai)</p> <p><<Interface>> IpInterface</p> <p>SIST ES 201 915-3 V1.4.1:2005 https://standards.iteh.ai/catalog/standards/sist/f7efbc9c-3d13-49f0-823c-6c291287dabb/sist-es-201-915-3-v1-4-1-2005</p>
--

5.3 Service Interfaces

5.3.1 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'.

5.4 Generic Service Interface

5.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.