

**Open Service Access (OSA);  
Application Programming Interface (API);  
Part 13: Policy Management SCF  
(Parlay 6)**

---



**ITeH STANDARD PREVIEW**  
(standards.iteh.ai)  
Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/68b1a4c4-7a3e-4407-a79c-3198609c509d/etsi-es-204-915-13-v1.1.1-2008-05>



## Reference

---

DES/TISPAN-01032-13-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](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.

**DECT**<sup>TM</sup>, **PLUGTESTS**<sup>TM</sup>, **UMTS**<sup>TM</sup>, **TIPHON**<sup>TM</sup>, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP**<sup>TM</sup> 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 Policy Management SCF.....	11
5 Sequence Diagrams .....	11
5.1 Use of Policy Repository.....	11
5.2 Introduce condition and action into rule.....	13
5.3 Create event.....	15
5.4 Create and modify domain .....	17
5.5 ASP offering services to prepaid subscribers .....	19
5.6 Create Signature for an evaluation context.....	21
5.7 Request Evaluation of Policies.....	23
5.8 Register for and Receive Notification of a Policy Event.....	23
6 Class Diagrams.....	24
6.1 PM Provisioning SCF Class Diagrams.....	25
6.2 PM Policy Evaluation SCF Class Diagrams.....	27
7 The Service Interface Specifications.....	27
7.1 Interface Specification Format .....	27
7.1.1 Interface Class .....	27
7.1.2 Method descriptions.....	27
7.1.3 Parameter descriptions.....	27
7.1.4 State Model.....	28
7.2 Base Interface .....	28
7.2.1 Interface Class IpInterface .....	28
7.3 Service Interfaces .....	28
7.3.1 Overview .....	28
7.4 Generic Service Interface .....	28
7.4.1 Interface Class IpService .....	28
7.4.1.1 Method setCallback().....	29
7.4.1.2 Method setCallbackWithSessionID().....	29
8 Policy Management (PM) Interface Classes .....	29
8.1 PM Provisioning SCF Interface Classes.....	29
8.1.1 Interface Class IpPolicyManager .....	30
8.1.1.1 Method createDomain() .....	30
8.1.1.2 Method getDomain().....	31
8.1.1.3 Method removeDomain().....	31
8.1.1.4 Method getDomainCount() .....	31
8.1.1.5 Method getDomainIterator() .....	31
8.1.1.6 Method findMatchingDomains().....	32
8.1.1.7 Method createRepository().....	32
8.1.1.8 Method getRepository() .....	32
8.1.1.9 Method removeRepository() .....	33
8.1.1.10 Method getRepositoryCount().....	33
8.1.1.11 Method getRepositoryIterator().....	33
8.1.1.12 Method startTransaction() .....	34
8.1.1.13 Method commitTransaction().....	34
8.1.1.14 Method abortTransaction().....	34

8.1.2	Interface Class IpPolicy .....	35
8.1.2.1	Attributes .....	35
8.1.2.2	Method getAttribute() .....	36
8.1.2.3	Method setAttribute() .....	36
8.1.2.4	Method getAttributes() .....	36
8.1.2.5	Method setAttributes() .....	37
8.1.3	Interface Class IpPolicyDomain .....	37
8.1.3.1	Attributes .....	39
8.1.3.2	Method getParentDomain() .....	40
8.1.3.3	Method createDomain() .....	40
8.1.3.4	Method getDomain() .....	40
8.1.3.5	Method removeDomain() .....	41
8.1.3.6	Method getDomainCount() .....	41
8.1.3.7	Method getDomainIterator() .....	41
8.1.3.8	Method createGroup() .....	41
8.1.3.9	Method getGroup() .....	42
8.1.3.10	Method removeGroup() .....	42
8.1.3.11	Method getGroupCount() .....	42
8.1.3.12	Method getGroupIterator() .....	43
8.1.3.13	Method createRule() .....	43
8.1.3.14	Method getRule() .....	43
8.1.3.15	Method removeRule() .....	44
8.1.3.16	Method getRuleCount() .....	44
8.1.3.17	Method getRuleIterator() .....	44
8.1.3.18	Method createEventDefinition() .....	44
8.1.3.19	Method getEventDefinition() .....	45
8.1.3.20	Method removeEventDefinition() .....	45
8.1.3.21	Method getEventDefinitionCount() .....	45
8.1.3.22	Method getEventDefinitionIterator() .....	46
8.1.3.23	Method createVariableSet() .....	46
8.1.3.24	Method getVariableSet() .....	46
8.1.3.25	Method removeVariableSet() .....	47
8.1.3.26	Method getVariableSetCount() .....	47
8.1.3.27	Method getVariableSetIterator() .....	47
8.1.3.28	Method createVariable() .....	47
8.1.3.29	Method setVariableValue() .....	48
8.1.3.30	Method getVariableType() .....	48
8.1.3.31	Method getVariableValue() .....	48
8.1.3.32	Method getVariable() .....	49
8.1.3.33	Method removeVariable() .....	49
8.1.3.34	Method createSignature() .....	50
8.1.3.35	Method getSignature() .....	50
8.1.3.36	Method removeSignature() .....	50
8.1.3.37	Method getSignatureCount() .....	50
8.1.3.38	Method getSignatureIterator() .....	51
8.1.4	Interface Class IpPolicyGroup .....	51
8.1.4.1	Attributes .....	52
8.1.4.2	Method getParentDomain() .....	53
8.1.4.3	Method getParentGroup() .....	53
8.1.4.4	Method createGroup() .....	53
8.1.4.5	Method getGroup() .....	54
8.1.4.6	Method removeGroup() .....	54
8.1.4.7	Method getGroupCount() .....	54
8.1.4.8	Method getGroupIterator() .....	54
8.1.4.9	Method createRule() .....	55
8.1.4.10	Method getRule() .....	55
8.1.4.11	Method removeRule() .....	55
8.1.4.12	Method getRuleCount() .....	56
8.1.4.13	Method getRuleIterator() .....	56
8.1.5	Interface Class IpPolicyRepository .....	56
8.1.5.1	Attributes .....	57
8.1.5.2	Method getParentRepository() .....	58

8.1.5.3	Method createRepository()	58
8.1.5.4	Method getRepository()	58
8.1.5.5	Method removeRepository()	59
8.1.5.6	Method getRepositoryCount()	59
8.1.5.7	Method getRepositoryIterator()	59
8.1.5.8	Method createCondition()	60
8.1.5.9	Method getCondition()	60
8.1.5.10	Method removeCondition()	60
8.1.5.11	Method getConditionCount()	61
8.1.5.12	Method getConditionIterator()	61
8.1.5.13	Method createAction()	61
8.1.5.14	Method getAction()	62
8.1.5.15	Method removeAction()	62
8.1.5.16	Method getActionCount()	62
8.1.5.17	Method getActionIterator()	62
8.1.6	Interface Class IpPolicyRule	63
8.1.6.1	Attributes	65
8.1.6.2	Method getParentGroup()	67
8.1.6.3	Method getParentDomain()	67
8.1.6.4	Method createCondition()	67
8.1.6.5	Method getCondition()	68
8.1.6.6	Method removeCondition()	68
8.1.6.7	Method getConditionCount()	68
8.1.6.8	Method getConditionIterator()	69
8.1.6.9	Method createAction()	69
8.1.6.10	Method getAction()	70
8.1.6.11	Method removeAction()	70
8.1.6.12	Method getActionCount()	70
8.1.6.13	Method getActionIterator()	70
8.1.6.14	Method setValidityPeriodConditionByName()	71
8.1.6.15	Method setValidityPeriodCondition()	71
8.1.6.16	Method getValidityPeriodCondition()	71
8.1.6.17	Method unsetValidityPeriodCondition()	71
8.1.6.18	Method setConditionList()	72
8.1.6.19	Method getConditionList()	72
8.1.6.20	Method setActionList()	72
8.1.6.21	Method getActionList()	72
8.1.7	Interface Class IpPolicyCondition	73
8.1.7.1	Attributes	74
8.1.7.2	Method getParentRepository()	75
8.1.7.3	Method getParentRule()	75
8.1.8	Interface Class IpPolicyTimePeriodCondition	75
8.1.8.1	Attributes	76
8.1.9	Interface Class IpPolicyAction	78
8.1.9.1	Attributes	79
8.1.9.2	Method getParentRepository()	80
8.1.9.3	Method getParentRule()	80
8.1.10	Interface Class IpPolicyEventDefinition	80
8.1.10.1	Attributes	81
8.1.10.2	Method setRequiredAttributes()	81
8.1.10.3	Method setOptionalAttributes()	82
8.1.10.4	Method getRequiredAttributes()	82
8.1.10.5	Method getOptionalAttributes()	82
8.1.10.6	Method getParentDomain()	82
8.1.11	Interface Class IpPolicyEventCondition	83
8.1.11.1	Attributes	83
8.1.12	Interface Class IpPolicyExpressionCondition	84
8.1.12.1	Attributes	84
8.1.13	Interface Class IpPolicyEventAction	85
8.1.13.1	Attributes	85
8.1.14	Interface Class IpPolicyExpressionAction	86
8.1.14.1	Attributes	86

8.1.15	Interface Class IpPolicyIterator .....	87
8.1.15.1	Attributes.....	87
8.1.15.2	Method getList() .....	88
8.1.16	Interface Class IpPolicySignature.....	88
8.1.16.1	Attributes.....	89
8.1.16.2	Method setInputVariables().....	90
8.1.16.3	Method setOutputVariables() .....	90
8.1.16.4	Method getInputVariables() .....	90
8.1.16.5	Method getOutputVariables() .....	91
8.1.16.6	Method setGroupNames() .....	91
8.1.16.7	Method setPolicyRoles() .....	91
8.1.16.8	Method getGroupNames() .....	91
8.1.16.9	Method getPolicyRoles() .....	92
8.1.16.10	Method getParentDomain().....	92
8.2	PM Policy Evaluation SCF Interface Classes.....	92
8.2.1	Interface Class IpPolicyEvalManager.....	93
8.2.1.1	Method evalPolicy().....	93
8.2.1.2	Method evalPolicyReq() .....	94
8.2.1.3	Method abortEvalPolicyReq().....	94
8.2.1.4	Method generateEvent().....	95
8.2.1.5	Method createNotification().....	95
8.2.1.6	Method destroyNotification().....	96
8.2.2	Interface Class IpAppPolicyDomain .....	96
8.2.2.1	Method reportNotification().....	96
8.2.2.2	Method evalPolicyRes().....	96
8.2.2.3	Method evalPolicyErr().....	97
9	State Transition Diagrams .....	97
9.1	PM Provisioning SCF State Transition Diagrams .....	97
9.2	PM Policy Evaluation SCF State Transition Diagrams .....	97
10	PM Service Properties .....	97
11	Data Definitions .....	98
11.1	Policy Management Data Definitions.....	98
11.1.1	TpPolicyConditionListType .....	98
11.1.2	TpPolicyConditionListElement .....	99
11.1.3	TpPolicyConditionList.....	99
11.1.4	TpPolicyConditionType.....	99
11.1.5	TpPolicyActionListElement .....	99
11.1.6	TpPolicyActionList.....	99
11.1.7	TpPolicyActionType.....	99
11.1.8	TpPolicyEvent .....	99
11.1.9	TpPolicyKeyword.....	100
11.1.10	TpPolicyKeywordSet.....	101
11.1.11	TpPolicyError .....	101
11.1.12	IpPolicyDomain .....	101
11.1.13	IpPolicyDomainRef .....	101
11.1.14	IpPolicyRepository .....	101
11.1.15	IpPolicyRepositoryRef.....	101
11.1.16	IpPolicyGroup.....	101
11.1.17	IpPolicyGroupRef.....	101
11.1.18	IpPolicyRule .....	101
11.1.19	IpPolicyRuleRef .....	102
11.1.20	IpPolicyEventDefinition .....	102
11.1.21	IpPolicyEventDefinitionRef .....	102
11.1.22	IpAppPolicyDomain .....	102
11.1.23	IpAppPolicyDomainRef .....	102
11.1.24	IpPolicyCondition.....	102
11.1.25	IpPolicyConditionRef .....	102
11.1.26	IpPolicyTimePeriodCondition .....	102
11.1.27	IpPolicyTimePeriodConditionRef .....	102
11.2	Data Types for PM Variables .....	102

11.2.1	TpPolicyVar.....	102
11.2.2	TpPolicyVarSet.....	103
11.2.3	TpPolicyRecordType.....	103
11.2.4	TpPolicyListType.....	103
11.2.5	TpPolicyTypeInfo.....	103
11.2.6	TpPolicyType.....	103
11.2.7	TpPolicyNameValue.....	104
11.2.8	TpPolicyNameValueList.....	104
11.3	eBNF for Condition and Action expressions.....	104
11.3.1	Basic Definition.....	104
11.3.2	Definitions of Constant (Literals).....	104
11.3.3	Definition of Operators.....	105
11.3.4	Allowable arithmetic expressions & predicates.....	105
11.3.5	Accessing Variables.....	105
11.3.6	Allowable Condition and Action Expressions.....	106
11.4	Example Scenarios.....	106
11.5	Example XML Scenarios.....	108
12	Policy Management Exception Classes.....	109
<b>Annex A (normative):</b>		
	<b>OMG IDL Description of Policy Management SCF.....</b>	<b>111</b>
<b>Annex B (informative):</b>		
	<b>W3C WSDL Description of the Policy Management SCF.....</b>	<b>112</b>
<b>Annex C (informative):</b>		
	<b>Java API Description of the Policy Management SCF.....</b>	<b>113</b>
<b>Annex D (informative):</b>		
	<b>Contents of 3GPP OSA R7 Policy Management.....</b>	<b>114</b>
<b>Annex E (informative):</b>		
	<b>Description of Policy Management for 3GPP2 cdma2000 networks.....</b>	<b>115</b>
E.1	General Exceptions.....	115
E.2	Specific Exceptions.....	115
E.2.1	Clause 1: Scope.....	115
E.2.2	Clause 2: References.....	115
E.2.3	Clause 3: Definitions and abbreviations.....	115
E.2.4	Clause 4: Policy Management SCF.....	115
E.2.5	Clause 5: Sequence Diagrams.....	115
E.2.6	Clause 6: Class Diagrams.....	115
E.2.7	Clause 7: The Service Interface Specifications.....	115
E.2.8	Clause 8: Policy Management Interface Classes.....	116
E.2.9	Clause 9: State Transition Diagrams.....	116
E.2.10	Clause 10: Data Definitions.....	116
E.2.11	Clause 11: Policy Management Exception Classes.....	116
E.2.12	Annex A (normative): OMG IDL Description of Policy Management SCF.....	116
<b>Annex F (informative):</b>		
	<b>Record of changes.....</b>	<b>117</b>
F.1	Interfaces.....	117
F.1.1	New.....	117
F.1.2	Deprecated.....	117
F.1.3	Removed.....	117
F.2	Methods.....	117
F.2.1	New.....	117
F.2.2	Deprecated.....	117
F.2.3	Modified.....	118
F.2.4	Removed.....	118
F.3	Data Definitions.....	118
F.3.1	New.....	118
F.3.2	Modified.....	118
F.3.3	Removed.....	118
F.4	Service Properties.....	118

F.4.1	New .....	118
F.4.2	Deprecated.....	119
F.4.3	Modified.....	119
F.4.4	Removed.....	119
F.5	Exceptions .....	119
F.5.1	New .....	119
F.5.2	Modified.....	119
F.5.3	Removed.....	119
F.6	Others .....	119
History	.....	120

**ITeH STANDARD PREVIEW**  
(standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/68b1a4c4-7a3e-4407-a79c-3198609c509d/etsi-es-204-915-13-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), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 13 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 29.198-13 V7.0.0 (Release 7).**

---

# 1 Scope

The present document is part 13 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 Policy Management Service Capability Feature (SCF) aspects of the interface. All aspects of the Policy Management 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).

---

## 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 Policy Management SCF

It is expected that more and more OSA services will use policies to express operational criteria. It is also expected that network providers will host policy-enabled services that have been written by 3<sup>rd</sup> party application service providers. In order to manage policy information, control access to it and to request evaluation of policies a policy management service is needed. Consistent with this, a policy management provisioning manager, IpPolicyManager, and a policy evaluation manager, IpPolicyEvalManager have been defined.

APIs have been defined to offer provisioning services. These include APIs to create, update or view policy information for any policy enabled service. Similarly APIs have been defined to facilitate interactions between clients (e.g. a 3<sup>rd</sup> party application) and the policies of any policy enabled service. These include APIs to subscribe to policy events, to request evaluation of policies and to request the generation of policy events. All APIs conform to an underlying policy information model that is derived from the policy core information model defined by the IETF in RFC 3460.

Clients that perform administrative tasks of behalf of a policy enabled service, e.g. create, update or delete policy information must obtain access to IpPolicyManager via the Framework. Administrative tasks may then be performed through methods supported by IpPolicyManager. Similarly, clients that need to invoke evaluation of policies of a specific policy enabled service may do so by obtaining access to IpPolicyEvalManager via the Framework.

Consistent with the above the Policy Management Service supports two classes of service interfaces for policy provisioning and policy evaluation. These are the PM Provisioning SCF and the PM Policy Evaluation SCF respectively.

Examples of policy enabled services include: A load balancing service that uses policies to manage application loads on the network, a charging service that determines charging criteria based on policies, a call management service that uses policies to direct end-user calls to appropriate call agents, etc.

Information in the present document is organized as follows:

- The Sequence diagrams give the reader a practical idea of how PM provisioning and PM evaluation SCFs are used by clients.
- The Class relationships clause shows relationships between the various interfaces supported by the PM provisioning and PM evaluation SCFs respectively.
- The Interface specification clauses describe in detail each of the interfaces shown within the Class diagram clause.
- 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.

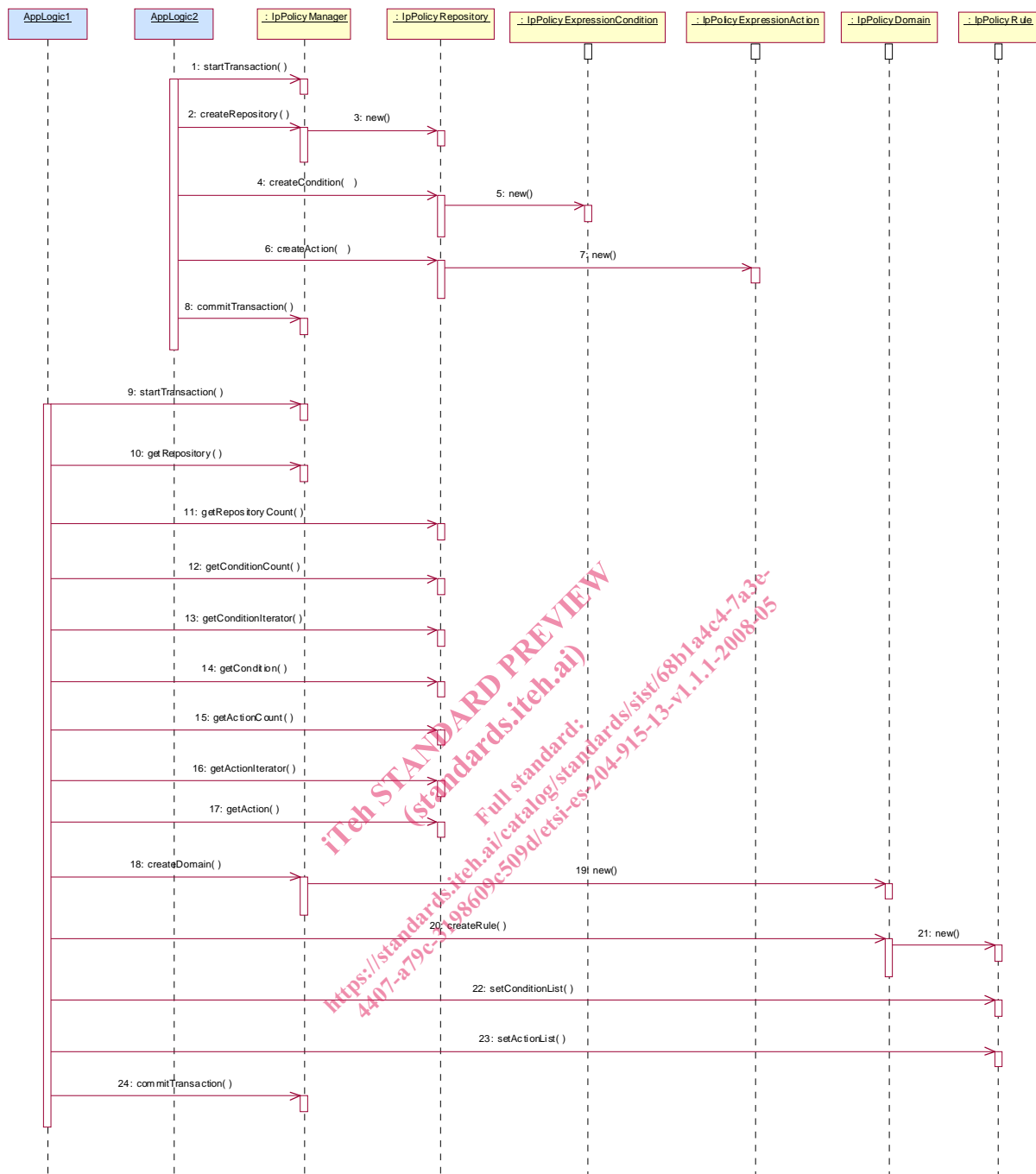
An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method. Where a method is not supported by an implementation of a Service interface, the exception P\_METHOD\_NOT\_SUPPORTED shall be returned to any call of that method.

---

## 5 Sequence Diagrams

### 5.1 Use of Policy Repository

The example shown here shows the use of a Policy Repository. The repository is meant to hold unattached conditions and actions. The Network Operator can populate the repository with the conditions and actions that it can support. These may indeed be based on 'off-line' negotiations with the application developer. The application developer uses the conditions and actions in the Policy Repository to create rules for his own application. In the example application logic represented by AppLogic1 belongs to the Network Operator, whereas the application logic represented by AppLogic2 belongs to the ASP. This example uses the same conditions, actions, and rules as the ASP example.



- 1: The creation of the repository by the Network Operator takes place within one transaction.
- 2: The method createRepository is invoked on the IpPolicyManager interface to create a new repository.
- 3: As a result of the createRepository method a new instance of the IpPolicyRepository interface is created. Its interface reference is returned as return parameter of the createRepository method.
- 4: The Network Operator creates an unattached condition in the new repository by invoking the createCondition method. For simplicity reasons, this is the same condition as in sequence 8 of the ASP example. The same condition attributes apply.
- 5: A new instance of the IpPolicyExpressionCondition interface is created.

- 6: The Network Operator creates an unattached action in the repository. Again, this is the same action as in sequence 10 of the ASP example. The same action attributes apply.
- 7: A new instance of the IpPolicyExpressionAction interface is created.
- 8: The Network Operator is finished with creating and populating the repository and closes the transaction.
- 9: Now that a repository exists, the ASP application can open a transaction to start creating a rule based on the conditions and actions stored in the repository.
- 10: The application invokes the getRepository to obtain a reference to the top-level repository. The returned reference in this case is the reference to the new repository just created by the Network Operator.
- 11: The application can invoke the getRepositoryCount method on the IpPolicyRepository interface to check whether there are any sub-repositories available. This is not the case for this example.
- 12: Before trying to obtain all available conditions in this repository the application retrieves the number of conditions by invoking the method getConditionCount.
- 13: The application can now invoke the getConditioniterator method to obtain the reference to an iterator that contains the names of each of the conditions contained by this repository that the application is authorized to see. As the previous method only return one available condition, this would be only one name, i.e. "SufficientCredit".
- 14: A reference to the condition can be obtained by invoking getCondition, with the condition name from the iterator as input parameter.
- 15: Similar to 12.
- 16: Similar to 13.
- 17: Similar to 14.
- 18: At this point in time the application has the names and references to the unattached condition and action from the repository it wants to use to create the rule. First a domain is created by invoking the createDomain method on the IpPolicyManager interface.
- 19: A new instance of the IpPolicyDomain interface is created.
- 20: The application invokes createRule to create a rule within the domain that was just created in flow 18 and 19.
- 21: A new instance of the IpPolicyRule interface is created.
- 22: By invoking the method setConditionList, the application can now apply the condition from the repository to this rule, by passing the condition reference, obtained by getCondition in flow 14, as an input parameter.
- 23: Similarly the application can apply the action to the rule by invoking setActionList.
- 24: Finally, once the rule is created using the condition and action from the policy repository, the transaction can be closed.

## 5.2 Introduce condition and action into rule

This sequence diagram describes how a specific policy rule is managed. A rule consists generally of conditions and of actions, the latter being evaluated if all conditions evaluate to true.

This sequence includes:

- creation of a condition and introduction of it into the rule;
- retrieval of an already defined action object from a repository and introduction into the rule;
- establishing a transaction bracket.

Presumption: the Application got a reference to the group, e.g. by having performed the sequence "create and modify domain" as in clause 5.4.