3GPP TS 28.311 V18.0.0 (2024-04)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Services and System Aspects;

Management and orchestration;

Network policy management for mobile networks based on Network Function Virtualization (NFV) scenarios

(Release 18)

** 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.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.

© 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered 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  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 6

1 Scope 8

2 References 8

3 Definitions of terms, symbols and abbreviations 8

3.1 Terms 8

3.2 Symbols 8

3.3 Abbreviations 8

4 Overview 9

5 Policy management architecture 9

6 Business level requirements 10

6.1 Requirements 10

6.2 Actor roles 10

6.3 Telecommunication resources 10

6.4 High-level use cases 10

6.4.1 Deploy the network function 10

7 Specification level requirements 10

7.1 Requirements 10

7.2 Actor roles 11

7.3 Telecommunications resources 11

7.4 Use cases 11

7.4.1 Create a policy in the context of NFV 11

7.4.2 Delete a policy in the context of NFV 12

7.4.3 Update a policy in the context of NFV 12

7.4.4 Query a policy in the context of NFV 13

7.4.5 Activate a policy in the context of NFV 13

7.4.6 Deactivate a policy in the context of NFV 14

7.4.7 Reporting policy conflicts in the context of NFV 14

8 Policy management procedures 14

8.1 General description for policy management procedures 14

8.2 Policy Creation 15

8.3 Policy Deletion 15

8.4 Policy Update 15

8.5 Policy Query 15

8.6 Policy Activation 16

8.7 Policy Deactivation 16

8.8 Policy Conflicts Notification 16

9 Information Object Classes 16

9.1 Information entities imported and local label 16

9.2 Class diagram 17

9.2.1 Attributes and relationships 17

9.2.2 Inheritance 17

9.3 Information object classes definition 17

9.3.1 Policy 17

9.3.1.1 Definition 17

9.3.1.2 Attributes 18

9.3.2 PolicyList 18

9.3.2.1 Definition 18

9.3.2.2 Attributes 18

9.3.3 PolicyManagementIRP 18

9.3.3.1 Definition 18

9.3.3.2 Attribute 18

9.3.3.3 Notifications 18

9.4 Information relationship definitions 18

9.4.1 relation-PolicyManagementIRP-PolicyList (M) 18

9.4.1.1 Definition 18

9.4.1.2 Role 18

9.4.1.3 Constraint 19

9.4.2 relation-PolicyList-Policy (M) 19

9.4.2.1 Definition 19

9.4.2.2 Role 19

9.4.2.3 Constraint 19

9.5 Information attribute definitions 19

9.5.0 Introduction 19

9.5.1 Definitions and legal values 20

10 Interface definition 20

10.1 Class diagram 20

10.2 PolicyManagementIRPOperations\_Interface (M) 21

10.2.1 Operation createPolicy (M) 21

10.2.1.1 Input parameters 21

10.2.1.2 Output parameters 21

10.2.1.3 Results 21

10.2.2 Operation deletePolicy (M) 21

10.2.2.1 Input parameters 21

10.2.2.2 Output parameters 21

10.2.2.3 Results 21

10.2.3 Operation updatePolicy (M) 22

10.2.3.1 Input parameters 22

10.2.2.2 Output parameters 22

10.2.2.3 Results 22

10.2.4 Operation queryPolicy (M) 22

10.2.4.1 Input parameters 22

10.2.4.2 Output parameters 22

10.2.4.3 Results 23

10.2.5 Operation activatePolicy (M) 23

10.2.5.1 Input parameters 23

10.3.5.2 Output parameters 23

10.3.5.3 Results 23

10.2.6 Operation deactivatePolicy (M) 23

10.2.6.1 Input parameters 23

10.2.6.2 Output parameters 23

10.2.6.3 Results 23

10.2.7 Operation queryPolicyList (M) 24

10.2.7.1 Input parameters 24

10.2.7.2 Output parameters 24

10.2.7.3 Results 24

10.3 PolicyManagementIRPNotification\_Interface (M) 24

10.3.1 policyConflictNotification 24

10.3.1.1 Input parameters 24

10.3.1.2 Triggering Event 24

10.3.1.2.1 From-state 24

10.3.1.2.2 To-state 24

11 Solution Set (SS) definitions 24

Annex A (normative): XML definitions 26

A.1 General 26

A.2 Architectural features 26

A.3 Mapping 26

A.3.1 General mapping 26

A.3.2 Information Object Class (IOC) mapping 26

A.4 Solution Set definitions 26

A.4.1 XML definition structure 26

A.4.2 Graphical Representation 26

A.4.3 XML Schema “PolicyManagement.xsd” 26

Annex B (normative): SOAP Solution Set 30

B.1 General 30

B.2 Architectural features 30

B.3 Mapping 30

B.3.1 Operation and notification mapping 30

B.3.2 Operation parameter mapping 30

B.3.2.0 General 30

B.3.2.1 Operation createPolicy 31

B.3.2.1.1 Input parameters 31

B.3.2.1.2 Output parameters 31

B.3.2.2 Operation deletePolicy 31

B.3.2.2.1 Input parameters 31

B.3.2.2.2 Output parameters 31

B.3.2.3 Operation updatePolicy 32

B.3.2.3.1 Input parameters 32

B.3.2.3.2 Output parameters 32

B.3.2.4 Operation queryPolicy 32

B.3.2.4.1 Input parameters 32

B.3.2.4.2 Output parameters 33

B.3.2.5 Operation activatePolicy 33

B.3.2.5.1 Input parameters 33

B.3.2.5.2 Output parameters 33

B.3.2.6 Operation deactivatePolicy 33

B.3.2.6.1 Input parameters 33

B.3.2.6.2 Output parameters 33

B.3.2.7 Operation queryPolicyList 34

B.3.2.7.1 Input parameters 34

B.3.2.7.2 Output parameters 34

B.3.2.8 Operation policyConflictNotification 34

B.3.2.8.1 Input parameters 34

B.3.2.8.2 Output parameters 34

B.4 Solution Set definitions 34

B.4.1 WSDL definition structure 34

B.4.2 Graphical Representation 34

B.4.3 WSDL specification “PolicyManagementIRPSystem.wsdl” 34

Annex C (informative): Change history 40

# 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.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document contains the architecture, requirements, use cases, procedures and definitions of interfaces for policy management for 4G networks.

# 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 TR 21.905: "Vocabulary for 3GPP Specifications".

[2] ETSI GR NFV-IFA 023 (V3.1.1): "Network Function Virtualisation (NFV); Management and Orchestration; Report on Policy Management in MANO; Release 3".

[3] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management: Information Service (IS)".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

Void.

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

EM Element Management

IRP Integration Reference Point

NE Network Element

NFVI Network Functions Virtualization Infrastructure

NFVO Network Functions Virtualization Orchestrator

NM Network Management

PAF Policy Administration Function

PF Policy Function

VIM Virtualized Infrastructure Manager

VNF Virtualized Network Function

VNFM VNF Manager

# 4 Overview

NFV policy management system may have policy conflict problem, which is a newly created policy may be conflict with an existing policy.

There are two potential policy conflict detection approaches, one is policy conflict detection in the Policy Function (PF), which makes the policy execution decisions, and the other one is policy conflict detection in the Policy Administration Function (PAF), which defines the network policy. There are multiple scenarios for policy conflict. For example, if the events are the same and the same condition are triggered (e.g. scaling policy on a VNF), but two policies have different actions, the conflict should be detected and resolved. It is not easy for one PF to resolve all of scenarios of the policy conflict. Hence, it is reasonable to enable PAF to resolve policy conflict, while the conflict detected by PAF or PF based on different specific scenarios, and the present document try to resolve the problem.

# 5 Policy management architecture

In order to delegate the support of policy control for virtualized NFs such as automatic scale in/out under certain condition , 3GPP management system would use the ETSI NFV defined related policy features/services. Consequently, two important logical functions are needed for the definition of the 3GPP policy management architecture.

1) Policy Administration Function (PAF) provides the services as follows:

- Allows the operator to define and administrate the network policy

2) Policy Function (PF) makes the decision for policy execution and provides the services as follows:

- Activating/deactivating the network policy

- Event subscription/notification of the network policy execution

- Providing relevant data and parameters during the process of policy execution (e.g. execution time consuming)

The 3GPP global policy management system would affect the behavior of all 3GPP defined nodes and would delegate the support of policy for virtualized NFs in ETSI MANO systems [2]. For NFV related policy management, 3GPP mainly focus on NM acting as PAF and EM acted as PF, and the interface between the NM and EM can be updated to support policy related operations. Hence, 3GPP policy management architecture under NFV scenario is as follows.



Figure 5-1: NFV policy management architecture in reference point presentation

NFV related policy interfaces between 3GPP management system and MANO such as Os-Ma-nfvo and Ve-Vnfm-em have been defined by ETSI.

# 6 Business level requirements

## 6.1 Requirements

**REQ-POM\_PO-CON-1** 3GPP management system should be able to support the capability about the network policy.

## 6.2 Actor roles

See detailed actors and roles for each use case in clause 6.4.

## 6.3 Telecommunication resources

See detailed telecommunication resources for each use case in clause 6.4.

## 6.4 High-level use cases

### 6.4.1 Deploy the network function

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | According to the network deployment policy, 3GPP system use the policy to deploy the corresponding network. |  |
| Actors and Roles | NM create/delete/update/query/activate/deactivate the network deployment policy, EM execute the NM’s policy operation request, and notifies NM policy conflict, if exist. |  |
| Telecom resources | 3GPP management system (EM, NM) |  |
| Assumptions | 3GPP management system want to deploy a network |  |
| Pre conditions | The operator designs the network deployment requirement and need deploy the network based on the requirement. |  |
| Begins when | There is a new network policy about the network deployment. |  |
| Step 1 (M) | The operator sets up the network deployment policy through NM. |  |
| Step 2 (M) | The NM activates the policy and notifies related EM. |  |
| Step 3 (M) | The EM follows the policy to design the network then the network is instantiated by the management system. |  |
| Ends when | The EM informs the NM that the policy has been executed. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The network is correctly configured and normally running |  |
| Traceability | REQ-POM\_PO-CON-1 |  |

# 7 Specification level requirements

## 7.1 Requirements

**REQ-POM-FUN-01**

The IRPAgent shall provide the capability to allow the IRPManager to create the network policy.

**REQ-POM-FUN-02**

The IRPAgent shall provide the capability to allow the IRPManager to delete the network policy.

**REQ-POM-FUN-03**

The IRPAgent shall provide the capability to allow the IRPManager to update the network policy.

**REQ-POM-FUN-04**

The IRPAgent shall provide the capability to allow the IRPManager to query the network policy.

**REQ-POM-FUN-05**

The IRPAgent shall provide the capability to allow the IRPManager to activate the network policy.

**REQ-POM-FUN-06**

The IRPAgent shall provide the capability to allow the IRPManager to deactivate the network policy.

**REQ-POM-FUN-07**

The IRPAgent should notify the IRPManager when there is a network policy conflict.

## 7.2 Actor roles

See detailed actors and roles for each use case in clause 7.4.

## 7.3 Telecommunications resources

See detailed telecommunication resources for each use case in clause 7.4.

## 7.4 Use cases

### 7.4.1 Create a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To create a policy such as deploying the SBC from IMS and P-GW from EPC in the same Data Centre. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. |  |
| Begins when | The IRPManager decides to create a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to create a network policy according to the network requirements, such as deploying the SBC. |  |
| Step 2 (M) | After the completion of the policy creation process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy is created and stored in the IRPAgent. |  |
| Traceability | REQ-POM-FUN -01 |  |

### 7.4.2 Delete a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To delete a policy such as deploying the SBC from IMS and P-GW from EPC in the same Data Center. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is a network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to delete a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to delete the existing policy. |  |
| Step 2 (M) | After the completion of the policy deletion process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy is deleted in the IRPAgent. |  |
| Traceability | REQ-POM-FUN -02 |  |

### 7.4.3 Update a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To update an existing network policy. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is a network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to update a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to update the existing policy. |  |
| Step 2 (M) | After the completion of the policy update process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy is updated and stored in the IRPAgent. |  |
| Traceability | REQ-POM-FUN -03 |  |

### 7.4.4 Query a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To query an existing network policy. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is a network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to query a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to query the existing policy. |  |
| Step 2 (M) | After the completion of the policy query process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy information is returned to the IPRManager. |  |
| Traceability | REQ-POM-FUN -04 |  |

### 7.4.5 Activate a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To activate an existing network policy. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is a network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to active a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to active the existing policy. |  |
| Step 2 (M) | After the completion of the policy activation process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy is activated in the IRPAgent. |  |
| Traceability | REQ-POM-FUN -05 |  |

### 7.4.6 Deactivate a policy in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To query an existing network policy. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is an active network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to deactive a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to deactive the existing policy. |  |
| Step 2 (M) | After the completion of the policy deactivation process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy is deactivated in the IRPAgent. |  |
| Traceability | REQ-POM-FUN -06 |  |

### 7.4.7 Reporting policy conflicts in the context of NFV

| **Use Case Stage** | **Evolution / Specification** | **<<Uses>>**  **Related use** |
| --- | --- | --- |
| Goal | To query an existing network policy. |  |
| Actors and Roles | IRPManager as user |  |
| Telecom resources | 3GPP management system (NM, EM). |  |
| Assumptions | 3GPP management systems are running normally, scenarios and policy requirements are clearly defined. |  |
| Pre conditions | The systems is correctly configured and normally running. There is a network policy stored in the IRPAgent. |  |
| Begins when | The IRPManager decides to update a policy. |  |
| Step 1 (M) | The IPRManager makes a decision to query the existing policy. |  |
| Step 2 (M) | After the completion of the policy conflicts notification process, the IRPAgent informs the IRPManager on the result of the process. |  |
| Ends when | Ends when all mandatory steps identified above are successfully completed or when an exception occurs. |  |
| Exceptions | One of the steps identified above fails. |  |
| Post Conditions | The policy conflicts notification is returned to the IRPManager. |  |
| Traceability | REQ-POM-FUN -07 |  |

# 8 Policy management procedures

## 8.1 General description for policy management procedures

There are two kinds of management procedures. The first kind includes those operations that allow the NM to invoke policy management operations towards the EM.

 Policy Creation

 Policy Deletion

 Policy Update

 Policy Query

 Policy Activation

 Policy Deactivation

The second kind includes the operation that the EM notifies NM policy conflicts found in EM.

 Policy Conflicts Notification.

## 8.2 Policy Creation

After the NM (IRP Manager) decides to create a policy (ies), it will send the create request to the EM (IRP Agent).



Figure 8.2: policy creation message flow

## 8.3 Policy Deletion

After the NM (IRP Manager) decides to delete a policy (ies), it will send the delete request to the EM (IRP Agent).



Figure 8.3: policy deletion message flow

## 8.4 Policy Update

After the NM (IRP Manager) decides to update a policy (ies), it will send the update request to the EM (IRP Agent).



Figure 8.4: policy update message flow

## 8.5 Policy Query

After the NM (IRP Manager) decides to query a policy, it will send the query request to the EM (IRP Agent). For the query request, the EM (IRP Agent) sends back the response with the policy to the NM (IRP Manager).



Figure 8.5: policy query message flow

## 8.6 Policy Activation

After the NM (IRP Manager) decides to activate a policy (ies), it will send the policy activation request to the EM (IRP Agent).



Figure 8.6: policy activation message flow

## 8.7 Policy Deactivation

After the NM (IRP Manager) decides to deactivate a policy (ies), it will send the policy deactivation request to the EM (IRP Agent).



Figure 8.7: policy deactivation message flow

## 8.8 Policy Conflicts Notification

When the EM (IRP Agent) receives new or updated policy (ies) from NM (IRP Manager), it will check whether the each received policy is conflicts with the previous storage policy. If the conflict is detected, the EM (IRP Agent) will notify the information to the NM (IRP Manager).



Figure 8.8: policy conflicts notification message flow

# 9 Information Object Classes

## 9.1 Information entities imported and local label

|  |  |
| --- | --- |
| Label reference | Local label |
| TS 32.312 [3], information object class, ManagedGenericIRP | ManagedGenericIRP |

## 9.2 Class diagram

### 9.2.1 Attributes and relationships

This clause introduces the set of information object classes (IOCs) that encapsulate capabilities contained within the IRP. This clause provides the overview of all support object classes in UML. Subsequent clauses provide more detailed specification of various aspects of these support object classes.

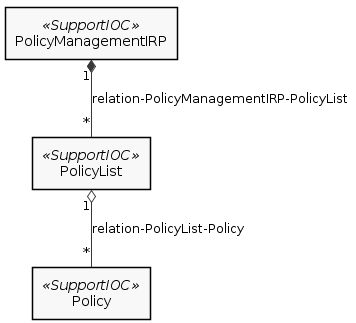


Figure 9.2.1: Information Object Class UML Diagram

### 9.2.2 Inheritance

This IOC inherits from ManagedGenericIRP IOC specified in TS.32.312 [3].

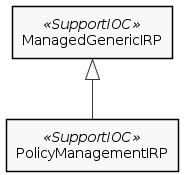


Figure 9.2.2: Information Object Class UML Diagram

## 9.3 Information object classes definition

### 9.3.1 Policy

#### 9.3.1.1 Definition

This IOC represents a network policy.

#### 9.3.1.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadble | isWriteble | isInvariant | isNotifyable |
| policyId | M | T | F | T | T |
| policyPriority | M | T | T | F | T |
| policyStatus | M | T | T | F | T |
| policyType | M | T | T | F | T |
| policyContent | M | T | T | F | T |

### 9.3.2 PolicyList

#### 9.3.2.1 Definition

PolicyList is the representation of the list of Policy.

#### 9.3.2.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadble | isWriteble | isInvariant | isNotifyable |
| policyListId | M | T | F | T | T |
| policyIdList | M | T | F | T | T |

### 9.3.3 PolicyManagementIRP

#### 9.3.3.1 Definition

PolicyManagementIRP is the representation of the policy management capabilities specified by the present document. This Support IOC inherits from ManagedGenericIRP Support IOC specified in 3GPP TS 32.312 [x].

#### 9.3.3.2 Attribute

There is no additional attribute defined for this IOC besides those inherited.

#### 9.3.3.3 Notifications

| Name | Qualifier | Notes |
| --- | --- | --- |
| policyConflictNotification | M | If there is network policy conflict(s), then the IRPAgent may send this notification to the IPRmanager |

## 9.4 Information relationship definitions

### 9.4.1 relation-PolicyManagementIRP-PolicyList (M)

#### 9.4.1.1 Definition

This represents the relationship between PolicyManagementIRP and PolicyList.

#### 9.4.1.2 Role

|  |  |
| --- | --- |
| Name | Definition |
| thePolicyList | It represents the PolicyList. |
| thePolicyManagementIRP | It represents the PolicyManagementIRP. |

#### 9.4.1.3 Constraint

|  |  |
| --- | --- |
| Name | Definition |
| uniquePolicyListId | The policyListIds playing the role of thePolicyList are unique within a particular PolicyManagementIRP. |

### 9.4.2 relation-PolicyList-Policy (M)

#### 9.4.2.1 Definition

This represents the relationship between PolicyList and Policy.

#### 9.4.2.2 Role

|  |  |
| --- | --- |
| Name | Definition |
| thePolicy | It represents the Policy. |
| thePolicyList | It represents the PolicyList. |

#### 9.4.2.3 Constraint

|  |  |
| --- | --- |
| Name | Definition |
| uniquePolicyId | The policyIds of all Policys, playing the role of the Policy, are unique within a particular PolicyManagementIRP and PolicyList. |

## 9.5 Information attribute definitions

### 9.5.0 Introduction

This clause defines the semantics of the attributes used in SupportIOCs.

### 9.5.1 Definitions and legal values

| Attribute Name | Definition | Legal Values |
| --- | --- | --- |
| policyId | It identifies the Policy instance (and distinguishes it from all other existing and stopped Policy instances of the PolicyManagementIRP Agent). | Any identifier. Value type is string. |
| policyPriority | It specifies the priority of Policy | Its value should be one of the following:  Low,  Medium,  High |
| policyStatus | It specifies the status of Policy. If a policy is activated, and then its status is active. If it is deactivated ,then its status is deactivated | Its value should be one of the following:  Activated,  Deactivated |
| policyType | It identifies a name of one policy type | Its value is network operator specific. Value type is string. |
| policyContent | It identifies the content of a network policy | Its value is network operator specific. Value type is string. |
| policyListId | It identifies the singleton PolicyList of the PolicyManagementIRP Agent. | Any identifier. Value type is string. |
| policyIdList | It is an array list of Policy.policyId. | Value type is array. |
| policyConflictNotification | It specifies which policies conflict. | Its value is vendor specific. Value type is string. |

# 10 Interface definition

## 10.1 Class diagram

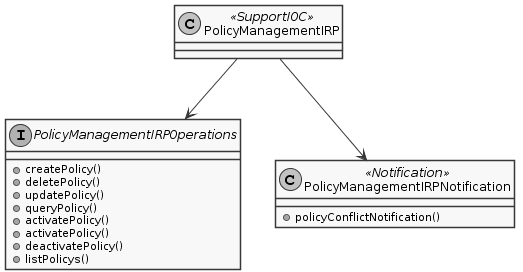


Figure 10.1: Class Diagram

## 10.2 PolicyManagementIRPOperations\_Interface (M)

### 10.2.1 Operation createPolicy (M)

#### 10.2.1.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.1.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values). |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.1.3 Results

In case of success, the corresponding policy information is created by the EM (IRP Agent). In case of failure, appropriate error information is returned.

### 10.2.2 Operation deletePolicy (M)

#### 10.2.2.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values). |

#### 10.2.2.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyId | M | Policy.policyId | The policyId of the deleted policy. See clause 9.5.1 (definitions and legal values). |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.2.3 Results

In case of success, the policy information are deleted by the EM (IRP Agent), and the corresponding deleted policy information is returned to the NM (IRP Manager). In case of failure, appropriate error information is returned.

### 10.2.3 Operation updatePolicy (M)

#### 10.2.3.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | O | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | O | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values).This parameter’s value is not allowed to be updated. |
| policyPriority | O | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). This parameter’s value is allowed to be updated. |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). This parameter’s value is allowed to be updated. |
| policyType | O | Policy.policyType | See clause 9.5.1 (definitions and legal values). This parameter’s value is allowed to be updated. |
| policyContent | O | Policy.policyContent | See clause 9.5.1 (definitions and legal values). This parameter’s value is allowed to be updated. |

#### 10.2.2.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyId | M | Policy.policyId | The policyId of the deleted policy. See clause 9.5.1 (definitions and legal values). |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.2.3 Results

In case of success, the corresponding policy information is updated by the EM (IRP Agent). In case of failure, appropriate error information is returned.

### 10.2.4 Operation queryPolicy (M)

#### 10.2.4.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values). |

#### 10.2.4.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| policyId | M | Policy.policyId | The policyId of the deleted policy. See clause 9.5.1 (definitions and legal values). |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.4.3 Results

In case of success, the corresponding policy information is returned by the EM (IRP Agent). In case of failure, appropriate error information is returned.

### 10.2.5 Operation activatePolicy (M)

#### 10.2.5.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values). |

#### 10.3.5.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| PolicyId | M | Policy.policyId | See clause 9.5.1 (definitions and legal values). Identifier of the activated Policy. |
| policyPriority | M | Policy.policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy.policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy.policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy.policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.3.5.3 Results

In case of success, the policy(ies) information are activated by the EM (IRP Agent) , and the activated policy information is returned to the NM (IRP Manager). In case of failure, appropriate error information is returned.

### 10.2.6 Operation deactivatePolicy (M)

#### 10.2.6.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyId | M | Policy. policyId | See clause 9.5.1 (definitions and legal values). |

#### 10.2.6.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| designer | M | Policy.designer | See clause 9.5.1 (definitions and legal values). |
| name | M | Policy.name | See clause 9.5.1 (definitions and legal values). |
| PolicyId | M | Policy. policyId | See clause 9.5.1 (definitions and legal values). Identifier of the activated Policy. |
| policyPriority | M | Policy. policyPriority | See clause 9.5.1 (definitions and legal values). |
| policyStatus | M | Policy. policyStatus | See clause 9.5.1 (definitions and legal values). |
| policyType | M | Policy. policyType | See clause 9.5.1 (definitions and legal values). |
| policyContent | M | Policy. policyContent | See clause 9.5.1 (definitions and legal values). |

#### 10.2.6.3 Results

In case of success, the policy(ies) information are deactivated by the EM (IRP Agent), and the deactivated policy information is returned to the NM (IRP Manager). In case of failure, appropriate error information is returned.

### 10.2.7 Operation queryPolicyList (M)

#### 10.2.7.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyListId | M | PolicyList.policyListId | See clause 9.5.1 (definitions and legal values). |

#### 10.2.7.2 Output parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| policyListId | M | PolicyList.policyListId | See clause 9.5.1 (definitions and legal values). |
| policyIdList | M | PolicyList.policyIdList | See clause 9.5.1 (definitions and legal values). |

#### 10.2.7.3 Results

In case of success, the EM (IRP Agent) returns the corresponding policy information. In case of failure, appropriate error information is returned.

## 10.3 PolicyManagementIRPNotification\_Interface (M)

### 10.3.1 policyConflictNotification

#### 10.3.1.1 Input parameters

| Parameter Name | Qualifier | Matching Information | Comment |
| --- | --- | --- | --- |
| activatedPolicyList | M | List of <PolicyList.policyListId> and their corresponding policy is activated. | See clause 9.5.1 (definitions and legal values). Identifier(s) of the activated Policy. |

#### 10.3.1.2 Triggering Event

##### 10.3.1.2.1 From-state

internalProblem

|  |  |
| --- | --- |
| Assertion Name | Definition |
| internalProblem | Because of a network policy conflict’s problem, PolicyManagementIRP Agent decides that it no longer can maintain the policy in any policyStatus but "deactivated". |

##### 10.3.1.2.2 To-state

policyConflictNotification

|  |  |
| --- | --- |
| Assertion Name | Definition |
| policyConflictNotification | There is network policyconflict(s). |

# 11 Solution Set (SS) definitions

The present document defines the following policy management Solution Set Definitions:

- Annex A provides the XML Definitions.

- Annex B provides the SOAP Solution Set.

Annex A (normative):  
XML definitions

# A.1 General

This annex contains the XML definitions for the policy management Integration Reference Point.

# A.2 Architectural features

The overall architectural feature of policy management IRP is specified, this clause specifies features that are specific to the Schema definitions.

The present document provides the main part of the XML file format definition for the policy management IRP IS.

The other parts of this XML definition are NRM-specific parts. All NRM IRPs that include SS-level XML definition are in the scope of the policy management IRP.

# A.3 Mapping

## A.3.1 General mapping

An IOC maps to an XML element of the same name as the IOC's name in the Information Model. An IOC attribute maps to a sub-element of the corresponding IOC's XML element, and the name of this sub-element is the same as the attribute's name in the Information Model.

## A.3.2 Information Object Class (IOC) mapping

The mapping is not present in the current version of the present document.

# A.4 Solution Set definitions

## A.4.1 XML definition structure

Clause A.4.2 provides a graphical representation of the XML elements.

Clause A.4.3 provides the schema fragment for policy management (policy creation/deletion/update/query/activation/deactivation/conflicts notification) XML files.

## A.4.2 Graphical Representation

The graphical representation is not present in the current version of the present document.

## A.4.3 XML Schema “PolicyManagement.xsd”

<?xml version="1.0" encoding="UTF-8"?>

<!--

3GPP TS 28.311 policy management XML Schema

PolicyOperation.xsd

-->

<schema xmlns:xb="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#PolicyManagement" xmlns:xe="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#PolicyManagement " xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#PolicyManagement" elementFormDefault="qualified" attributeFormDefault="unqualified">

<!—createPolicy Request -->

<element name="createPolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- createPolicy Response -->

<element name=" createPolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- deletePolicy Request -->

<element name="deletePolicy">

<complexType>

<sequence>

<element name="policyId" type="string"/>

</sequence>

</complexType>

</element>

<!-- deletePolicy Response -->

<element name="deletePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- updatePolicy Request -->

<element name="updatePolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- updatePolicy Response -->

<element name="updatePolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- queryPolicy Request -->

<element name="queryPolicy">

<complexType>

<sequence>

<element name="policyId " type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- queryPolicy Response -->

<element name="queryPolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- activatePolicy Request -->

<element name="activatePolicy">

<complexType>

<sequence>

<element name="policyId " type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- activatePolicy Response -->

<element name=" activatePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- deactivatePolicy Request -->

<element name="deactivatePolicy">

<complexType>

<sequence>

<element name="policyId" type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- deactivatePolicy Response -->

<element name=" deactivatePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- queryPolicyList Request -->

<element name="queryPolicyList">

<complexType>

<sequence>

<element name="policyListId" type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- queryPolicyList Response -->

<element name="queryPolicyList">

<complexType>

<sequence>

<element name="policyListId" type="Identifier"/>

<element name="policyIdList" type="array"/>

</sequence>

</complexType>

</element>

<!-- policyConflictNotification Request -->

<element name="policyConflictNotification">

<complexType>

<sequence>

<element name="activatedPolicyList" type="Identifier"/>

</sequence>

</complexType>

</element>

</schema>

Annex B (normative):   
SOAP Solution Set

# B.1 General

This annex contains the XML definitions for the policy management Integration Reference Point.

# B.2 Architectural features

The overall architectural feature of the policy management IRP is specified in the present document. This clause specifies features that are specific to the SOAP Solution Set.

The present document uses "document" style in the WSDL description.

The present document uses "literal" encoding style in the WSDL description.

# B.3 Mapping

## B.3.1 Operation and notification mapping

Policy management IRP management: Clause 9 defines semantics of operation visible across the Itf-N. Table B.3 indicates mapping of these operations to their equivalents defined in this SS.

Table B.3: Mapping from IS Operation to SS Equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation | SS Operation | Qualifier |
| createPolicy | createPolicy | M |
| deletePolicy | deletePolicy | M |
| updatePolicy | updatePolicy | M |
| queryPolicy | queryPolicy | M |
| activatePolicy | activatePolicy | M |
| deactivatePolicy | deactivatePolicy | M |
| queryPolicyList | queryPolicyList | M |
| policyConflictNotification | policyConflictNotification | M |

## B.3.2 Operation parameter mapping

### B.3.2.0 General

The clause 10 defines semantics of parameters carried in operations. The tables below show the mapping of these parameters, as per operation, to their equivalents defined in this SS.

### B.3.2.1 Operation createPolicy

#### B.3.2.1.1 Input parameters

Table B.3.2.1.1: Mapping from IS createPolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

#### B.3.2.1.2 Output parameters

Table B.3.2.1.2: Mapping from IS createPolicy output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.2 Operation deletePolicy

#### B.3.2.2.1 Input parameters

Table B.3.2.2.1: Mapping from IS deletePolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyId | policyInfoId | M |

#### B.3.2.2.2 Output parameters

Table B.3.2.2.2: Mapping from IS deletePolicy output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.3 Operation updatePolicy

#### B.3.2.3.1 Input parameters

Table B.3.2.3.1: Mapping from IS updatePolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | O |
| name | name | O |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | O |
| policyStatus | policyStatus | M |
| policyType | policyType | O |
| policyContent | policyContent | O |

#### B.3.2.3.2 Output parameters

Table B.3.2.3.2: Mapping from IS updatePolicy output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.4 Operation queryPolicy

#### B.3.2.4.1 Input parameters

Table B.3.2.4.1: Mapping from IS queryPolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyId | policyInfoId | M |

#### B.3.2.4.2 Output parameters

Table B.3.2.4.2: Mapping from IS queryPolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.5 Operation activatePolicy

#### B.3.2.5.1 Input parameters

Table B.3.2.5.1: Mapping from IS activatePolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyId | policyInfoId | M |

#### B.3.2.5.2 Output parameters

Table B.3.2.5.2: Mapping from IS startSession input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.6 Operation deactivatePolicy

#### B.3.2.6.1 Input parameters

Table B.3.2.6.1: Mapping from IS deactivatePolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyId | policyInfoId | M |

#### B.3.2.6.2 Output parameters

Table B.3.2.6.2: Mapping from IS deactivatePolicy input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| designer | designer | M |
| name | name | M |
| policyId | policyInfoId | M |
| policyPriority | policyPriority | M |
| policyStatus | policyStatus | M |
| policyType | policyType | M |
| policyContent | policyContent | M |

### B.3.2.7 Operation queryPolicyList

#### B.3.2.7.1 Input parameters

Table B.3.2.7.1: Mapping from IS listPolicys input parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyListId | policyListId | M |

#### B.3.2.7.2 Output parameters

Table B.3.2.7.2: Mapping from IS listPolicys output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| policyListId | policyListId | M |
| policyIdList | policyIdList | M |

### B.3.2.8 Operation policyConflictNotification

#### B.3.2.8.1 Input parameters

Table B.3.2.8.1: Mapping from IS policyConflictNOtificaition output parameters to SS equivalents

|  |  |  |
| --- | --- | --- |
| IS Operation parameter | SS Operation parameter | Qualifier |
| activatedPolicyList | activatedPolicyList | M |

#### B.3.2.8.2 Output parameters

Void.

# B.4 Solution Set definitions

## B.4.1 WSDL definition structure

The present document defines the main part of what are supported by the policy management IRP agent.

## B.4.2 Graphical Representation

The graphical representation is not present in the current version of the present document.

## B.4.3 WSDL specification “PolicyManagementIRPSystem.wsdl”

<?xml version="1.0" encoding="UTF-8"?>

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:PolicyManagement IRPSystem=http://www.3gpp.org/ftp/Specs/archive/28\_series/28.311/schema/28311/PolicyManagementIRPSystem

xmlns:xs="http://www.w3.org/2001/XMLSchema"

targetNamespace="http://www.3gpp.org/ftp/Specs/archive/28\_series/28.311/schema/28311/GenericIRPSystem">

<types>

<!—createPolicy Request -->

<element name="createPolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- createPolicy Response -->

<element name=" createPolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- deletePolicy Request -->

<element name="deletePolicy">

<complexType>

<sequence>

<element name="policyId" type="string"/>

</sequence>

</complexType>

</element>

<!-- deletePolicy Response -->

<element name="deletePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- updatePolicy Request -->

<element name="updatePolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- updatePolicy Response -->

<element name="updatePolicy">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- queryPolicy Request -->

<element name="queryPolicy">

<complexType>

<sequence>

<element name="policyId " type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- queryPolicy Response -->

<element name="queryPolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- activatePolicy Request -->

<element name="activatePolicy">

<complexType>

<sequence>

<element name="policyId " type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- activatePolicy Response -->

<element name=" activatePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- deactivatePolicy Request -->

<element name="deactivatePolicy">

<complexType>

<sequence>

<element name="policyId " type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- deactivatePolicy Response -->

<element name=" deactivatePolicyResponse">

<complexType>

<sequence>

<element name="designer" type="string"/>

<element name="name" type="string"/>

<element name="policyId " type="Identifier"/>

<element name="policyPriority" type="string"/>

<element name="policyStatus" type="string"/>

<element name="policyType" type="string"/>

<element name="policyContent" type="string"/>

</sequence>

</complexType>

</element>

<!-- queryPolicyList Request -->

<element name="queryPolicyList">

<complexType>

<sequence>

<element name="policyListId" type="Identifier"/>

</sequence>

</complexType>

</element>

<!-- queryPolicyList Response -->

<element name="queryPolicyList">

<complexType>

<sequence>

<element name="policyListId" type="Identifier"/>

<element name="policyIdList" type="array"/>

</sequence>

</complexType>

</element>

<!-- policyConflictNotification Request -->

<element name="policyConflictNotification">

<complexType>

<sequence>

<element name="activatedPolicyList" type="Identifier"/>

</sequence>

</complexType>

</element>

</types>

<message name="createPolicy Request">

<part name="parameter" element="createPolicy"/>

</message>

<message name="createPolicy Response">

<part name="parameter" element="createPolicyResponse"/>

</message>

<message name="deletePolicy Request ">

<part name="parameter" element="deletePolicy"/>

</message>

<message name="deletePolicy Response">

<part name="parameter" element="deletePolicyResponse"/>

</message>

<message name="updatePolicy Request ">

<part name="parameter" element="PolicyUpdate"/>

</message>

<message name="updatePolicy Response">

<part name="parameter" element="updatePolicyResponse"/>

</message>

<message name="queryPolicyRequest ">

<part name="parameter" element="queryPolicy"/>

</message>

<message name="queryPolicyResponse">

<part name="parameter" element="queryPolicyResponse"/>

</message>

<message name="activePolicyRequest ">

<part name="parameter" element="ActivePolicy"/>

</message>

<message name="activePolicyResponse">

<part name="parameter" element="ActivePolicyResponse"/>

</message>

<message name="deactivePolicyRequest ">

<part name="parameter" element="deactivePolicy"/>

</message>

<message name="deactivePolicyResponse">

<part name="parameter" element="deactivePolicyResponse"/>

</message>

<message name="queryPolicyListRequest">

<part name="parameter" element="queryPolicyList"/>

</message>

<message name="queryPolicyListResponse">

<part name="parameter" element="queryPolicyListResponse"/>

</message>

<message name="policyConflictNotificaitonRequest">

<part name="parameter" element="policyConflictsNotificaiton"/>

</message>

<portType name="PolicyMangementIRPPortType">

<operation name="createPolicy">

<input message="PolicyMangementIRPSystem:createPolicyRequest"/>

<output message="PolicyMangementIRPSystem:createPolicyResponse"/>

</operation>

<operation name="deletePolicy">

<input message="PolicyMangementIRPSystem:deletePolicyRequest"/>

<output message="PolicyMangementIRPSystem:deletePolicyResponse"/>

</operation>

<operation name="updatePolicy">

<input message="PolicyMangementIRPSystem:updatePolicyRequest"/>

<output message="PolicyMangementIRPSystem:updatePolicyResponse"/>

</operation>

<operation name="queryPolicy">

<input message="PolicyQueryIRPSystem:queryPolicyRequest"/>

<output message="PolicyQueryIRPSystem:queryPolicyResponse"/>

</operation>

<operation name="activePolicy">

<input message="PolicyMangementIRPSystem:activePolicyRequest"/>

<output message="PolicyMangementIRPSystem:activePolicyResponse"/>

</operation>

<operation name="deactivePolicy">

<input message="PolicyMangementIRPSystem:deactivePolicyRequest"/>

<output message="PolicyMangementIRPSystem:deactivePolicyResponse"/>

</operation>

<operation name="queryPolicyList">

<input message="PolicyMangementIRPSystem:queryPolicyListRequest"/>

<output message="PolicyMangementIRPSystem:queryPolicyListResponse"/>

</operation>

<operation name="PolicyConflictNotification">

<input message="PolicyMangementIRPSystem:policyConflictNotificationRequest"/>

</operation>

</portType>

<binding name="PolicyManagementIRPBinding" type="PolicyManagementIRPSystem: PolicyManagementIRPPortType">

<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="createPolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# createPolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="deletePolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# deletePolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="updatePolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# updatePolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="queryPolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# queryPolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="activePolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# activePolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="deactivePolicy">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# deactivePolicy" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="queryPolicyList">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311# queryPolicyList" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

<operation name="policyConflictNofitication">

<soap:operation soapAction="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#PolicyConflictNofitication" style="document"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

</binding>

<service name="PolicyMangementIRPService">

<port name="PolicyMangementIRPPort" binding="PolicyMangementIRPSystem: PolicyMangementIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#BulkCMIRP"/>

</port>

<port name="GenericIRPPort" binding="genericIRPSystem:GenericIRPBinding">

<soap:address location="http://www.3gpp.org/ftp/specs/archive/28\_series/28.311#GenericIRP"/>

</port>

</service>

</definitions>

Annex C (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2018-09 |  |  |  |  |  | Skeleton | 0.0.0 |
| 2018-10 | SA5#121 | S5-186398  S5-186468 |  |  |  | Update to implement the agreed pCRs in SA5#121:  S5-186398  pCR 28.311 Add Scope  S5-186468 Rel-16 draft TS Skeleton 28.311 | 0.1.0 |
| 2018-11 | SA5#122 | S5-187385  S5-187386 |  |  |  | Update to implement the agreed pCRs in SA5#122:  S5-187385  Add Policy management architecture  S5-187386 Add Business level requirements | 0.2.0 |
| 2019-06 | SA5#125Adhoc | S5-194428  S5-194429  S5-194430  S5-194431  S5-194432  S5-194433  S5-194434  S5-194435 |  |  |  | Update to implement the agreed pCRs in SA5#125Adhoc:  S5-194428 Policy Management Procedures  S5-194429 Policy Creation  S5-194430 Policy Deletion  S5-194431 Policy Update  S5-194432 Policy Query  S5-194433 Policy Activation  S5-194434 Policy Deactivation  S5-194435 Policy Conflicts Notification | 0.3.0 |
| 2019-08 | SA5#126 | S5-195179  S5-195843  S5-195845  S5-195846  S5-195847 |  |  |  | Update to implement the agreed pCRs in SA5#126:  S5-195179 Add abbreviations  S5-195843 Add references  S5-195845 Add overview  S5-195846 Rewrite business level requirements  S5-195847 Add specification level requirements | 0.4.0 |
| 2019-10 | SA5#127 | S5-196658 |  |  |  | Update to implement the agreed pCRs in SA5#127:  S5-196658 Add Solution Set(SS) definitions | 0.5.0 |
| 2019-11 | SA5#128 | S5-197763 S5-197764 S5-197765  S5-197766 |  |  |  | Update to implement the agreed pCRs in SA5#128:  S5-197763 pCR 28.311 Add Information Object Classes  S5-197764 pCR 28.311 Add XML definition  S5-197765 pCR 28.311 Add SOAP Solution Set  S5-197766 pCR 28.311 Rewrite interface | 0.6.0 |
| 2019-12 | SA#86 | SP-191185 |  |  |  | Presented for information and approval | 1.0.0 |
| 2019-12 | SA#86 |  |  |  |  | Change control version | 16.0.0 |
| 2022-03 | - | - | - | - | - | Update to Rel-17 version (MCC) | **17.0.0** |
| 2024-04 | - | - | - | - | - | Update to Rel-18 version (MCC) | **18.0.0** |