

# Configuration Guide for ADC Based on ADC Rules (Sd)

Ericsson Service-Aware Policy Controller

User Guide

## **Copyright**

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

## **Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

## **Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document [Trademark Information](#).



# Contents

<b>1</b>	<b>Configure ADC Based on ADC Rules (Sd) Overview</b>	<b>1</b>
1.1	Typographic Conventions	2
1.2	Other Conventions	2
<b>2</b>	<b>Configuration Prerequisites</b>	<b>5</b>
<b>3</b>	<b>Sd Diameter Data</b>	<b>6</b>
3.1	Configure Gx PCEF Nodes	6
3.2	Configure Sd TDF Nodes	7
3.3	Configure TDF Selection	7
3.3.1	Select TDF Conditionally	8
3.3.2	Select TDF Unconditionally	8
3.4	Configure Event Triggers	9
3.5	Provision Services for ADC Rules	9
3.5.1	Set ADC rules	10
3.6	Configure Service Access Control	11
3.6.1	Provision Policies for Service Authorization	12
3.6.2	Provision Static Service Policies	13
3.7	Configure Service Charging Control	15
3.8	Provision Dynamic Services	16
3.9	Configure Dynamic Service Classification for Sd	16
3.10	Configure Dynamic Service Qualification in Sd	20
3.10.1	Configure Static Qualification for Dynamic Services in Sd	20
3.10.2	Configure Dynamic Qualification for Dynamic Services in Sd	20
3.11	Configure Bearer QoS Control for Dynamic Services	23
3.12	Configure Access and Charging Policies Based on Dynamic Service Establishment	24
<b>4</b>	<b>Configuration Examples for Use Cases</b>	<b>25</b>
4.1	Upgrade Default Bearer to Prioritize Streaming Delivery	25
4.2	Allocate Dedicated Bearer to Prioritize Streaming Delivery	26
<b>5</b>	<b>Appendix A. ADC over Sd Policy Types</b>	<b>28</b>
<b>6</b>	<b>Appendix B. ADC over Sd Policy Tags</b>	<b>30</b>
<b>7</b>	<b>Reference List</b>	<b>31</b>





# 1 Configure ADC Based on ADC Rules (Sd) Overview

Figure 1 shows the main parts related to configuring and provisioning the SAPC.

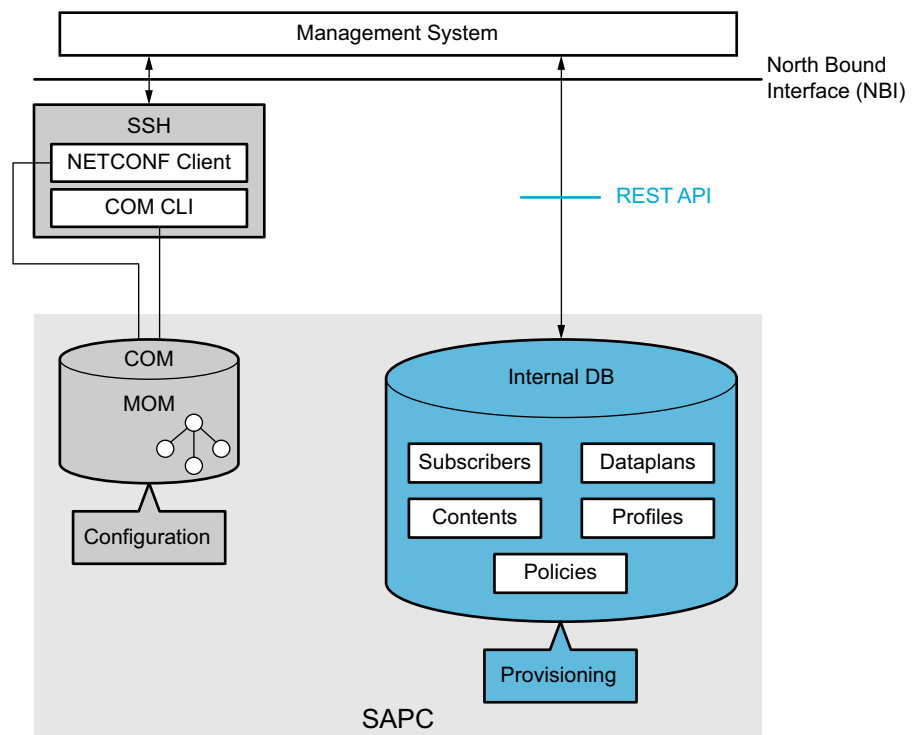


Figure 1 Configuration and Provisioning Overview

The purpose of this document is to provide guidelines to configure the SAPC for Application Detection and Control (ADC) based on ADC rules and Dynamic Policy Control over the Sd interface by providing configuration examples.

This document does not intend to be a complete guide for configuring all possibilities related to ADC rules and Dynamic Policy Control over Sd in the SAPC.

The complete parameter list and details of all configured options of the SAPC are included in separate documents, refer to Managed Object Model (MOM) and Provisioning REST API.

For general concepts about the provisioning of policies, see Configuration Guide for Subscription and Policies.

**Note:** The configuration for ADC based on PCC rules (Gx) is already covered in Configuration Guide for ADC based on PCC rules (Gx).



Examples in this document cover the case of data configured in the SAPC internal repository. If an external repository is used, refer to [Database Access](#).

## 1.1 Typographic Conventions

The following typographic and document conventions are used:

Table 1 Typographic Conventions

Convention	Description	Example
<b>Representational State Transfer (REST)</b>	SAPC REST provisioning.  Exact REST resources, methods, attributes, or their corresponding values.	<code>PUT /dataplan/Silver { "dataplanName" : "Silver", "notification" : "sms" }</code>
<b>Managed Object Class (MOC) or Attributes value</b>	Exact COM model object, classes names, attributes, or their corresponding values.	<code>SmsCenter enableDelivery=true</code>
<b>NETCONF</b>	SAPC COM configuration	<pre>&lt;edit-config&gt; &lt;target&gt; &lt;running /&gt; &lt;/target&gt; &lt;config&gt; &lt;ManagedElement xmlns="urn:com:ericsson:ecim:ComTop"&gt; &lt;managedElementId&gt; 1 &lt;/managedElementId&gt; &lt;PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom"&gt; &lt;policyControlFunctionId&gt; 1 &lt;/ policyControlFunctionId&gt; &lt;NotificationConfig xmlns="urn:com:ericsson: ecim:notificationconfigmom"&gt; &lt;notificationConfigId&gt; 1 &lt;/ notificationConfigId&gt; &lt;enableDelivery&gt; true &lt;/ enableDelivery&gt; &lt;/NotificationConfig&gt; &lt;/ PolicyControlFunction&gt; &lt;/ManagedElement&gt; &lt;/ config&gt; &lt;/edit-config&gt;</pre>

## 1.2 Other Conventions

This document refers to some configuration and provisioning data.

To clarify which detailed data is managed by COM or by the REST API, this document uses the following conventions:



- Configuration: whenever referring to Managed Object Class (MOC).

The detailed description of the object and attributes can be found in Managed Object Model (MOM).

Example: set enableReauthsOnSubsChange attribute in class AppConfig.

The tools or interfaces to manage these data in the SAPC are:

- NETCONF interface, refer to [Ericsson NETCONF Interface](#).

The configuration examples show the NETCONF file contents, using the following syntax:

```
<edit-config>
...
<config>
  <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
    <managedElementId>1</managedElementId>
    ...
  </ManagedElement>
</config>
</edit-config>
```

- COM CLI, refer to [Ericsson Command-Line Interface](#).

- Provisioning: mainly subscribers, subscriber groups (dataplan), services (contents), profiles, and policy-related data. The SAPC provides a REST API for them, see [Provisioning REST API](#).

This document uses the following terminology for them: <resource-name> URI in the provisioning REST API.

Example: To provision subscriber groups, use the dataplan URI in the provisioning REST API.

Provisioning examples show HTTP operations on REST resources with the following syntax:

```
HTTP-Operation /resource-URI
{json content}
```

where /resource-URI is the relative URI from the SAPC provisioning base URI detailed in [Provisioning REST API](#).

Example:

```
PUT /dataplan/Gold
{ "dataplanName" : "Gold",
  "subscribedContents" : [{ "contentName" : "HTTP_Streaming",
```



```
        "redirect" : false}]  
    }
```

**Note:** To ease provisioning operations, the SAPC provides an HTTPS CLI client named `resty`, refer to [Provisioning Tools](#).





## 2 Configuration Prerequisites

Before configuring the SAPC in an operational network, assure that:

- CBA Components are installed.
- The SAPC product software is installed.
- To have a detailed understanding of the function.

## 3 Sd Diameter Data

To support ADC and Dynamic Policy Control over Sd in the SAPC, configure the PCEF and Traffic Detection Function (TDF) node data to be able to send and receive messages over the Gx and Sd interfaces:

- [Configure Gx PCEF nodes](#)
- [Configure Sd TDF nodes](#)
- [Configure TDF selection](#)

**Note:** The Origin-Host, Origin-Realm, IP address and diameter port values are set during the SAPC installation procedure. Diameter data related to capabilities exchange (application and vendor identifiers) are provided at installation time, so that no manual procedure is needed.

### 3.1 Configure Gx PCEF Nodes

An Sd session towards a TDF can be established per IP-CAN session only if the PCEF that initiates the IP-CAN session does not support ADC based on PCC rules enhanced with ADC.

To support an Sd session for an IP-CAN session initiated from a PCEF, make sure that the `adcSupport` attribute is not set to "true" at `DiameterNode` level.

The PCEF configuration to support Dynamic Policy Control in Sd is the same configuration needed to support Dynamic Policy Control in Rx. For further details refer to the [Dynamic Policy Control \(Rx\)](#) and [Configuration Guide for Dynamic Policy Control \(Rx\)](#) documents.

The procedure to configure the set of policy controls that the SAPC applies for a PCEF node is detailed in [Configuration Guide for Access and Charging Control \(Gx\)](#). The Dynamic Policy Control over Sd refers to the following controls:

- To enable support for dynamic PCC Rules in the PCEF based on notifications received over the Sd set `dynamicServiceSupport` attribute to "true" at `DiameterNode` level.
- Additionally, set the following policy controls related to Sd dynamic services:
  - `BEARER_QOS`, to allocate QoS information to Sd dynamic services
  - `SERVICE_CHARGING`, to allocate charging information to Sd dynamic services

**Note:** In multiple Gx scenarios where an IP-CAN session is controlled by several PCEFs, the Sd session is associated with only one Gx session.



## 3.2 Configure Sd TDF Nodes

When the SAPC establishes an Sd session towards a TDF, the TDF connection details can be received over the Gx interface during the Gx session establishment or can be previously configured in the SAPC.

A TDF is configured in the SAPC including the Diameter realm and Diameter host or the TDF IP address, as shown in [Example 1](#).

### Example 1 Sd TDF Node Configuration

```
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
      <managedElementId>1</managedElementId>
      <PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom">
        <policyControlFunctionId>1</policyControlFunctionId>
        <Network xmlns="urn:com:ericsson:ecim:networkmom">
          <networkId>1</networkId>
          <DiameterNodes>
            <diameterNodesId>1</diameterNodesId>
            <Tdfs>
              <tdfsId>1</tdfsId>
              <Tdf>
                <tdfId>tdf1</tdfId>
                <host>TDFNodeHostname1.tdf1gpprealm.com</host>
                <realm>tdf1gpprealm.com</realm>
              </Tdf>
              <Tdf>
                <tdfId>tdf2</tdfId>
                <ipAddress>192.168.14.42</ipAddress>
              </Tdf>
              <Tdf>
                <tdfId>tdf3</tdfId>
                <host>TDFNodeHostname3.tdf3gpprealm.com</host>
                <realm>tdf3gpprealm.com</realm>
                <ipAddress>192.168.14.43</ipAddress>
              </Tdf>
            </Tdfs>
          </DiameterNodes>
        </Network>
      </PolicyControlFunction>
    </ManagedElement>
  </config>
</edit-config>
```

**Note:** When the Diameter host, Diameter realm, and TDF IP addresses are provisioned, the SAPC only uses the host and realm to establish the Diameter session with the TDF.

## 3.3 Configure TDF Selection

Once the SAPC identifies that an Sd session has to be established, and in case the TDF diameter routing details are not provided during the Gx session establishment, the SAPC needs to find the TDF intended to provide ADC to the PCEF that is requesting the IP-CAN session establishment.



### 3.3.1 Select TDF Conditionally

To assign a TDF to a PCEF conditionally, create a `TdfRoute` object at `DiameterNode` level. In the `tdfId` attributeset a valid TDF name as already defined in [Configure Sd TDF Nodes](#) on page 7.

The `apns` attribute includes the list of APNs for which the indicated TDF must be used for ADC, as shown in [Example 2](#).

#### Example 2 Conditional TDF Selection

```
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
      <managedElementId>1</managedElementId>
      <PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom">
        <policyControlFunctionId>1</policyControlFunctionId>
        <Network xmlns="urn:com:ericsson:ecim:networkmom">
          <networkId>1</networkId>
          <DiameterNodes>
            <diameterNodesId>1</diameterNodesId>
            <DiameterNode>
              <diameterNodeId>PCEF1</diameterNodeId>
              <controls>IP_CAN_SESSION_ACCESS</controls>
              <controls>SERVICE_ACCESS_PCEF_TOD</controls>
              <controls>BEARER_QOS</controls>
              <controls>CONTENT_FILTERING</controls>
              <dynamicServiceSupport>true</dynamicServiceSupport>
              <adcSupport>false</adcSupport>
              <TdfRoute>
                <tdfId>tdf1</tdfId>
                <apns>APN1,APN5,APN9</apns>
              </TdfRoute>
            </DiameterNode>
          </DiameterNodes>
        </Network>
      </PolicyControlFunction>
    </ManagedElement>
  </config>
</edit-config>
```

If the `tdfId` attribute contains a nonexistent or wrong TDF name, the SAPC cannot establish the Sd session.

An APN should not be configured in more than one `TdfRoute`. Otherwise, it is considered an erroneous configuration. The SAPC manages this scenario by selecting the first TDF found for the target APN as the TDF to be used for the Sd session establishment.

### 3.3.2 Select TDF Unconditionally

To assign a TDF to a PCEF unconditionally (also known as default TDF), configure a `TdfRoute` object at `DiameterNode` level. In the `tdfId` attribute set a valid TDF name that is already defined in [Configure Sd TDF Nodes](#) on page 7. The `apns` attribute is an empty value. TDF conditionally selected has precedence over the default. See [Example 3](#) for further details.



### Example 3 Unconditional TDF Selection

```
<edit-config>
  <target>
    <running/>
  </target>
  <config>
    <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
      <managedElementId>1</managedElementId>
      <PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom">
        <policyControlFunctionId>1</policyControlFunctionId>
        <Network xmlns="urn:com:ericsson:ecim:networkmom">
          <networkId>1</networkId>
          <DiameterNodes>
            <diameterNodesId>1</diameterNodesId>
            <DiameterNode>
              <diameterNodeId>PCEF2</diameterNodeId>
              <controls>IP_CAN_SESSION_ACCESS</controls>
              <controls>SERVICE_ACCESS_PCEF_TOD</controls>
              <controls>BEARER_QOS</controls>
              <controls>CONTENT_FILTERING</controls>
              <dynamicServiceSupport>true</dynamicServiceSupport>
              <adcSupport>false</adcSupport>
              <TdfRoute>
                <tdfId>tdf2</tdfId>
                <apns></apns>
              </TdfRoute>
            </DiameterNode>
          </DiameterNodes>
        </Network>
      </PolicyControlFunction>
    </ManagedElement>
  </config>
</edit-config>
```

If the `tdfId` attribute contains a nonexistent or wrong TDF name, the SAPC cannot establish the Sd session.

The configuration of more than one TDFs selected unconditionally per PCEF is considered an erroneous configuration. The SAPC manages this scenario by selecting the first TDF found as the default TDF per corresponding Diameter node.

**Note:** To configure a TDF for Clustered Diameter Systems, it is also necessary to add the `TdfRoute` object under the corresponding `DiameterNode` MOC with the `clusterPattern` attribute set to the proper value.

## 3.4 Configure Event Triggers

To indicate to the TDF that the SAPC is about to start or stop application reporting, the SAPC always sends `APPLICATION_START` and `APPLICATION_STOP` event trigger values to the TDF during the Sd session establishment.

There is no need to configure any `eventTriggers` attribute at subscriber, dataplan, or SAPC level.

## 3.5 Provision Services for ADC Rules

The services controlled by the SAPC over the Sd interface must be provisioned. To provision services use the contents URI in the provisioning REST API.



The SAPC only supports the following type of services in the Sd interface:

- **Static:** predefined in the TDF, activated by the SAPC, and identified by the ADC-Rule-Name AVP or the ADC-Rule-Base-Name AVP in Sd interface.

### 3.5.1 Set ADC rules

To set an ADC rule for a static service, fill the following attributes in the corresponding contents URI in the provisioning REST API:

#### Steps

1. Fill the `adcRuleName` (unique identifier) attribute
2. Set the `adcRuleType` attribute to:
  - When the ADC rule is identified by name, set the `adcRuleType` attribute to "0"
  - When the ADC rule is identified by basename, set the `adcRuleType` attribute to "1"

**Note:** The name and basename must be the same as provisioned in the TDF.

[Example 4](#) shows the provisioning of different services and ADC rules.

#### Example 4 Provisioning of Services

```
PUT /contents/Chat
{
  "contentName" : "Chat",
  "adcRuleName" : "1000",
  "adcRuleType" : 0
}

PUT /contents/Internet
{
  "contentName" : "Internet",
  "adcRuleName" : "2000",
  "adcRuleType" : 1
}

PUT /contents/Skype
{
  "contentName" : "Skype",
  "pccRuleName" : "8001",
  "pccRuleType" : 0,
  "adcRuleName" : "8002",
  "adcRuleType" : 0
}
```



```
}

```

This example provisions the following services:

- "Chat" service

Static service known by the TDF. A static ADC rule identified by the `adcRuleName` (when `adcRuleType` is set to 0, the ADC-Rule-Name AVP is sent to the TDF).

- "Internet" service

Static service known by the TDF. A static ADC rule identified by the `adcRuleName` (when `adcRuleType` is set to 1, the ADC-Rule-Base-Name AVP is sent to the TDF).

- "Skype" (Voice over IP) service

Static service known by both the TDF and the PCEF. A static ADC rule identified by the `adcRuleName` (`adcRuleType` 0) is sent to the TDF or a static PCC rule identified by the `pccRuleName` (`pccRuleType` 0) is sent to the

**Note:** When a PCC rule is provisioned in the same content as an ADC rule, the SAPC downloads the PCC rule to the PCEF only if that PCEF supports ADC function over the Gx interface.

## 3.6 Configure Service Access Control

Service Access Control applied over a content containing ADC rules determines the services to be monitored on a particular Sd session. It happens in a similar way that Service Access Control applied over a content containing PCC rules determines the services to be authorized for a particular Gx session.

The services over Sd applicable to a subscriber must be known by the SAPC. These services must be provisioned for the subscriber or the dataplan in the `subscribedContents` attribute of the subscriber or dataplan. Refer to [Configuration Guide for Access and Charging Control \(Gx\)](#) for more information.

[Table 2](#) shows the policy types related to service access, which can be used and configured in the SAPC for service access over Sd.

Table 2 Sd Service Access Related Policies

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Access Control (Service Authorization)	access	<contentId>	<subscriberId> <dataplanId>		Sd Conditions:



Access					—Subscriber —Access Data —SubsCharging —Tdf Data ToD
Access Control (Static Service Qualification) <b>Static Access</b>	static-access	<contentId>	<subscriberId> <dataplanId>	permit <b>adc-rule-id</b> "<adcRuleName>"	<b>Sd</b> Conditions: —Subscriber —Access Data —SubsCharging —Tdf Data ToD

### 3.6.1 Provision Policies for Service Authorization

To configure Service Authorization depending on Conditions, create the necessary policies using the following URIs in the provisioning REST API:

- For **Global policy locator**:

/locators/resources/<contentName>/contexts/access

- For **Subscriber group locator**:

/dataplanes/<dataplanName>/locators/resources/<contentName>/contexts/access

- For **Subscriber locator**:

/subscribers/<subscriberId>/locators/resources/<contentName>/contexts/access

**Note:** It is not necessary to use the outputAttributes object.

**Note:** Non-authorization codes are not supported over the Sd interface.

In [Example 5](#) the "Internet" service is authorized for the subscriber if access is performed through IP-CAN type 5 (3GPP-EPS). If the condition is not fulfilled, the service is not authorized. Therefore, the corresponding ADC rule is not downloaded to the TDF over the Sd interface.

#### Example 5 Configuration of Service Authorization for Internet Service

```
PUT /rules/SAuth_Internet_AccessBased
{
  "condition" : "(AccessData.bearer.ipCanType==5)",
  "ruleName" : "SAuth_Internet_AccessBased"
}
```





```

PUT /policies/SAuth_Internet_Policy_1
{
  "policyName" : "SAuth_Internet_Policy_1",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "SAuth_Internet_AccessBased" ]
}

PUT /subscribers/3460000001/locators/resources/Internet/contexts/access
{
  "policies" : [ "SAuth_Internet_Policy_1" ]
}

PUT /subscribers/3460000001
{
  "subscriberId" : "3460000001",
  "subscribedContents" :
  [
    {
      "contentName" : "Internet"
    }
  ]
}

```

### 3.6.2 Provision Static Service Policies

Together with the TDF, the SAPC can assign and change the bandwidth limits and rating group for static services during an Sd session lifetime, depending on flexible conditions for a subscriber or subscriber group.

Some examples of possible use cases are the following:

- Two different ratings can be applied to data services in the Home Network or in roaming for premium and professional subscribers
- Throttle the bandwidth to 128 Kbps for P2P file sharing in peak hours

A static service in the TDF can be identified by several static service names. Each static service name in the TDF assigns different characteristics (rating group, service bandwidth, and so on) to the service. The SAPC authorizes the static base name taking into account subscriber information, accumulated use, and so on. The SAPC then selects the right `AdcRuleName` in terms of bandwidth and rating group, depending on the conditions configured in the policies.

To configure Static Services depending on Conditions, create the needed policies using:

- For **Global policy locator**:

```
/locators/resources/<contentName>/contexts/static-access
```

- For **Subscriber group locator**

```
/dataplan/<dataplanName>/locators/resources/<contentName>/
contexts/static-access
```

- For **Subscriber locator**



```
/subscribers/<subscriberId>/locators/resources/<contentName>/  
contexts/static-access
```

- Within the outputAttributes object in the rule set:
  - attrName attribute to adc-rule-id.
  - attrValue attribute to the new adcRuleName (ADC-Rule-Name value).

Table 3 ADC-Rule-Base-Name and ADC-Rule-Name Relation

Service	adcRuleName (ADC-Rule-Base-Name AVP)	adcRuleType	Condition	new adcRuleName (ADC-Rule-Name AVP)
P2P	20	1 (basename)	Not roaming Roaming	20001 20002

Table 4 TDF Local Data Associated with Downloaded ADC-Rule-Name Values

ADC-Rule-Name	Bandwidth	RatingGroup
20001	1 Mbps	Flat rate
20002	128 Kbps	Not flat rate

In [Example 6](#), for the "P2P" service, depending whether the subscriber is in roaming or not, a different bandwidth and rating group is applied, by sending either ADC-Rule-Name = 20002 or ADC-Rule-Name = 20001 to the TDF.

Example 6 Configuration for Qualifying Static ADC Rules for TDF

```
# -----  
PUT /rules/rQualify1  
{  
  "condition" : "AccessData.subscriber.locationInfo.countryCode == 214",  
  "outputAttributes" :  
  [  
    {  
      "attrName" : "adc-rule-id",  
      "attrValue" : "\"20001\"",  
      "result" : "permit"  
    }  
  ],  
  "ruleName" : "rQualify1"  
}  
  
PUT /rules/rQualify2  
{  
  "condition" : "not(AccessData.subscriber.locationInfo.countryCode == 214)",  
  "outputAttributes" :  
  [  
    {  
      "attrName" : "adc-rule-id",  
      "attrValue" : "\"20002\"",  
      "result" : "permit"  
    }  
  ],  
  "ruleName" : "rQualify2"  
}  
  
PUT /policies/pQualify  
{  
  "policyName" : "pQualify",
```



```

    "ruleCombiningAlgorithm" : "permit-overrides",
    "rules" : [ "rQualify1", "rQualify2" ]
  }
PUT /dataplan/OneGroup/locators/resources/P2P/contexts/static-access
{
  "policies" : [ "pQualify" ]
}
PUT /contents/P2P
{
  "contentName" : "P2P",
  "adcRuleName" : "20",
  "adcRuleType" : 1
}
PUT /dataplan/OneGroup
{
  "dataplanName" : "OneGroup",
  "subscribedContents" :
  [
    {
      "contentName" : "P2P",
      "redirect" : false
    }
  ]
}

```

Bandwidth limits and rating group for static defined services can also be changed depending on time conditions. A specific bandwidth and rating group can be applied in flat hours, and a different bandwidth and rating group for the rest of the time. To achieve that, modify the condition attribute of the rules in the previous example. For example:

- condition for ADC-Rule-Name = 20002:

```

"condition" : "(now.time > "08:00:00") && (now.time < "18:00:01")"

```

- condition for ADC-Rule-Name = 20001:

```

"condition" : "(now.time < "8:00:01") || (now.time > "18:00:00")"

```

## 3.7 Configure Service Charging Control

To select different charging data for static services (ADC rules) in the TDF, the SAPC allows the following options:

- Unconditionally: provision adcRuleName in the content resource
- Conditionally: configure static-access polices as explained in [Provision Static Service Policies](#) on page 13



## 3.8 Provision Dynamic Services

In addition to the dynamic services activated in the SAPC based on the information coming from the AF (using the Rx interface), the SAPC supports dynamic services activated in the SAPC based on the information coming from the TDF (using the Sd interface) and dynamically provisioned towards the GGSN/PDN-GW.

For further information regarding the provisioning of dynamic services, refer to the [Dynamic Policy Control \(Rx\)](#) and [Configuration Guide for Dynamic Policy Control \(Rx\)](#) documents.

## 3.9 Configure Dynamic Service Classification for Sd

The SAPC needs to determine a service identifier corresponding to the dynamic service activated by TDF events. The service identifier is relevant to perform dynamic service qualification afterwards. For further information on how to configure the conditions to determine such a service identifier, see [Configuration Guide for Dynamic Policy Control \(Rx\)](#).

The SAPC uses the following information received over the Sd interface for Dynamic Service Classification:

- TDF application identifier
- TDF application instance identifier
- Server IP address
- Server port

The output result is the identifier of the service. The association between dynamic services and the information provided by the TDF is flexible, and requires a detailed knowledge about the TDF-Application-Identifier or TDF-Application-Instance-Identifier data, or both that trigger the activation of dynamic services in the SAPC.

To provision policies for Dynamic Service Classification for Sd, use the values summarized in the following table:

Table 5 Dynamic Service Classification Policies for Sd

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Dynamic Service Classification for Sd	service-classification	tdf-detected-application	-	<b>service</b> <contentId>	Algorithms: — any-match  Conditions:



					— Tdf Data
--	--	--	--	--	------------

The result of service classification in Sd is successful if the SAPC matches the application detection information (which includes the TDF application identifier, TDF application instance identifier, and the flow information, provided in the CC-Request Update command from the TDF) with any of the configured rules in the applicable policies:

- Use the `/locators/resources/tdf-detected-application/contexts/service-classification` URI in the provisioning REST API containing:

- In the `policies` attribute, the names of the policies involved in the classification

- For each policy, use the `policies` URI in the provisioning REST API:

- Choose `any-match` as the `ruleCombiningAlgorithm` attribute.

The `any-match` attribute means that the result of the policy contains the output attribute of all the rules included in the policy with the "true" result.

- In the `rules` attribute, add the names of the rules for this policy.

- For each rule, use the `rules` URI in the provisioning REST API:

- In the `condition` attribute, include the TDF application identifier, the TDF application instance identifier, the server IP address and/or the server port used to identify the service. The SAPC uses this information to determine whether an evaluated application detection information matches the rule or not. Use the expression language detailed in [Configuration Guide for Subscription and Policies](#). See [appendix B](#) for a complete list of the policy tags that can be used.
- In the `attrName` attribute of the `outputAttributes` object, include the "service" value
- In the `attrValue` attribute of the `outputAttributes` object, include the service identifier result of Dynamic Service Classification

When several rules in a policy are met for different application detection information, the process returns the list of service identifiers of all matched rules.

**Note:** Only global policy locator policies can be configured for Dynamic Service Classification.

[Table 6](#) shows examples for dynamic service classification for Sd, whose columns contain the input and output information elements. The rows in the table are evaluated in order.

Table 6 Dynamic Service Classification in Sd

TDF-Application-Identifier	TDF-Application-Instance-Identifier	Condition	Service Id
"skype"	"audio"	application="skype", instance="audio"	Skype_audio
"skype"	"video"	application="skype", instance="video"	Skype_video
"facebook"	"streaming"	application="facebook", server port="995"	Facebook_streaming

Figure 2 shows the required configuration for two services ("Skype\_audio" and "Skype\_video") in one policy and for the "Facebook\_streaming" service in a different policy:

**Note:** The "Skype" (audio and video) and "Facebook" services can also be configured using one policy only, since the any-match attribute does not impose any restrictions on the number of rules configured per policy.

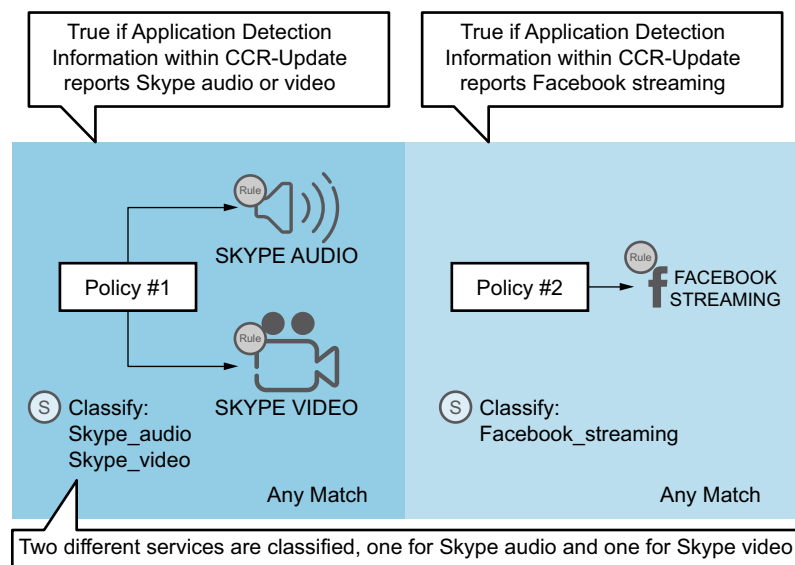


Figure 2 Example of Configuration to Classify the Skype Audio and Video and Facebook Streaming Services

Example 7 shows the configuration corresponding to the services and conditions described in Table 6:

Example 7 Configuration of Classification Policies for Dynamic Services in Sd

```
# -----
PUT /rules/classifySkypeAudio
{
  "condition" : "(TdfData.tdfAppId == \"skype\")&&(TdfData.tdfAppInstanceId ==
  \"audio\")",
  "outputAttributes" :
  [
    {
      "attrName" : "service",
```



```

        "attrValue" : "\"Skype_audio\"",
        "result" : "permit"
    },
    ],
    "ruleName" : "classifySkypeAudio"
}
PUT /rules/classifySkypeVideo
{
    "condition" : "(TdfData.tdfAppId == \"skype\")&&(TdfData.tdfAppInstanceId == →
    \"video\")",
    "outputAttributes" :
    [
        {
            "attrName" : "service",
            "attrValue" : "\"Skype_video\"",
            "result" : "permit"
        }
    ],
    "ruleName" : "classifySkypeVideo"
}
PUT /policies/pClassifySkype
{
    "policyName" : "pClassifySkype",
    "ruleCombiningAlgorithm" : "any-match",
    "rules" : [ "classifySkypeAudio", "classifySkypeVideo" ]
}
PUT /rules/classifyFacebookStreaming
{
    "condition" : "(TdfData.tdfAppId == \"facebook\")&&(TdfData.port == \"995\") →
    ",
    "outputAttributes" :
    [
        {
            "attrName" : "service",
            "attrValue" : "\"Facebook_streaming\"",
            "result" : "permit"
        }
    ],
    "ruleName" : "classifyFacebookStreaming"
}
PUT /policies/pClassifyFacebook
{
    "policyName" : "pClassifyFacebook",
    "ruleCombiningAlgorithm" : "any-match",
    "rules" : [ "classifyFacebookStreaming" ]
}
PUT /locators/resources/tdf-detected-application/contexts/service-classification →
{
    "policies" : [ "pClassifySkype", "pClassifyFacebook" ]
}

```

This configuration example results in the following classification for dynamic services depending on the data received in the Sd requests:

- "Skype\_video" and "Skype\_audio": in this case, any application detection information containing the TDF application identifier with "Skype" value and TDF application instance identifier with "audio" or "video" value will match the policy
- "Facebook\_streaming": in this case, any application detection information containing the 995 remote server port and the TDF-Application-Identifier AVP with "Facebook" value will match the policy

Once a dynamic service is classified, a charging profile or a QoS profile, or both can be associated with it, using the dynamic service qualification [Configure](#)



[Dynamic Service Qualification in Sd](#) on page 20process. For further details, see [Configure Dynamic Service Qualification in Sd](#) on page 20.

## 3.10 Configure Dynamic Service Qualification in Sd

Dynamic services over Sd can be qualified with QoS or charging data either statically or using policy conditions. The SAPC first evaluates the service qualification policies, but if there are not applicable policies, or the policies are not fulfilled, the SAPC obtains the QoS and charging profiles statically assigned to the dynamic service.

### 3.10.1 Configure Static Qualification for Dynamic Services in Sd

The association of a static QoS profile or a static charging profile with a dynamic service follows the same process as for preconfigured services:

- For QoS profile qualification, refer to [Configuration Guide for Bearer QoS and Bandwidth Management](#)
- For charging profile qualification, refer to [Configuration Guide for Access and Charging Control \(Gx\)](#)

[Example 8](#) shows the allocation of static QoS and charging profiles to a dynamic service:

**Example 8** Static QoS and Charging Profiles Assigned to Dynamic Services

```
PUT /contents/Skype_video
{
  "contentName" : "Skype_video",
  "staticQualification" :
  {
    "contentChargingProfileId" : "Char_Skype_video",
    "contentQosProfileId" : "QoS_Skype_video"
  }
}

PUT /profiles/content-qos/QoS_Skype_video
{
  "arpPriorityLevel" : 7,
  "profileId" : "QoS_Skype_video",
  "qci" : 2
}

PUT /profiles/content-charging/Char_Skype_video
{
  "chargingServiceId" : 301,
  "profileId" : "Char_Skype_video",
  "ratingGroup" : 301,
  "reportingLevel" : 1
}
```

### 3.10.2 Configure Dynamic Qualification for Dynamic Services in Sd

Using policies, it is possible to use dynamic conditions to assign QoS and charging profiles to dynamic services in Sd.





To use policies for dynamic service qualification, create the needed policies using the following URIs in the provisioning REST API:

- For **global policy locator**:

```
/locators/resources/<contentName>/contexts/<context type>
```

- For **subscriber group locator**:

```
/dataplan/<dataplanName>/locators/resources/<contentName>/  
contexts/<context type>
```

- For **subscriber locator**:

```
/subscribers/<subscriberId>/locators/resources/<contentName>/  
contexts/<context type>
```

Where <contentName> is the given name of the service to be qualified and <context type> is either charging or qos.

In addition, to define the conditions that need to be fulfilled to apply a given QoS or charging profile, define qualification rules and qualification policies by using the following tables:

Table 7 Configure Dynamic Service Qualification in Sd

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Bearer QoS Control (Dynamic Service Qualification) <b>QoS for Service</b>	qos	<contentId>	<subscriberId> <dataplanId>	permit <b>qos</b> ServiceQoSProfile ["<qosProfileName>"]	<b>Mixing policies and qualification Conditions:</b>  —Subscriber —Access Data —SubsCharging —Tdf Data  ToD
(Dynamic Service Qualification) <b>Service Charging</b>	charging	<contentId>	<subscriberId> <dataplanId>	permit <b>charging</b> ServiceChargingProfile ["<chargingProfileName>"]	<b>Mixing policies and qualification Conditions:</b>  —Subscriber —Access Data —SubsCharging —Tdf Data  ToD



[Example 9](#) shows how to assign a QoS profile depending on policy conditions to the "Skype\_video" dynamic service:

#### Example 9 Configuration of QoS Upgrade for Dynamic Services

```
PUT /profiles/content-qos/Qos_Upgrade
{
  "arpPci" : false,
  "arpPriorityLevel" : 5,
  "arpPvi" : true,
  "gbrDownlink" : 2000,
  "gbrUplink" : 500,
  "mbrDownlink" : 2500,
  "mbrUplink" : 1000,
  "profileId" : "Qos_Upgrade"
}

PUT /rules/qualifySkype_video
{
  "condition" : "(AccessData.bearer.accessPoint == \"internet.network2.operatorX\")",
  "outputAttributes" :
  [
    {
      "attrName" : "qos",
      "attrValue" : "ServiceQosProfile[\"Qos_Upgrade\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "qualifySkype_video"
}

PUT /policies/pQualifySkype_video
{
  "policyName" : "pQualifySkype_video",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "qualifySkype_video" ]
}

PUT /locators/resources/Skype_video/contexts/qos
{
  "policies" : [ "pQualifySkype_video" ]
}
```

[Example 9](#) shows how to configure a QoS profile named "Qos\_Upgrade" for the "Skype\_audio" dynamic service when the APN is internet.network2.operatorX.

#### Example 10 Configuration of Charging Data for Dynamic Services

[Example 10](#) shows how to configure a charging data profile for dynamic services depending on conditions:

```
PUT /profiles/content-charging/ChargingType1
{
  "meteringMethod" : 1,
  "offlineEnabled" : true,
  "onlineEnabled" : false,
  "profileId" : "ChargingType1",
  "ratingGroup" : 4,
  "reportingLevel" : 1
}

PUT /rules/qualifyFacebook_streaming
{
  "condition" : "(AccessData.subscriber.locationInfo.countryCode == \"214\") & & (AccessData.subscriber.locationInfo.networkCode == \"07\")",
  "outputAttributes" :
  [
```



```

        {
            "attrName" : "charging",
            "attrValue" : "ServiceChargingProfile[\"ChargingType1\"]",
            "result" : "permit"
        }
    ],
    "ruleName" : "qualifyFacebook_streaming"
}
PUT /policies/pQualifyFacebook_streaming
{
    "policyName" : "pQualifyFacebook_streaming",
    "ruleCombiningAlgorithm" : "permit-overrides",
    "rules" : [ "qualifyFacebook_streaming" ]
}
PUT /locators/resources/Facebook_streaming/contexts/charging
{
    "policies" : [ "pQualifyFacebook_streaming" ]
}

```

The “ChargingType1” charging profile is configured for the “Facebook\_streaming” dynamic service when the user location information indicates that MCC=214 and MNC=07.

### 3.11 Configure Bearer QoS Control for Dynamic Services

In scenarios where dynamic services do not require a dedicated bearer, but are installed on the default IP-CAN session bearer, it may be required to configure the QoS of the default bearer in connection with the establishment of dynamic services on the IP-CAN session.

The configuration of bearer QoS control is described in [Configuration Guide for Bearer QoS and Bandwidth Management](#). In addition, the following specific QoS selection tags are defined in connection with dynamic services based on notifications received from the Rx and Sd interfaces:

- `maxDynQosProfile` that returns a QoS profile composed of the highest value for every field in the QoS profile, out of the values obtained for each dynamic PCC Rule running in the IP-CAN session
- `sumDynQosProfile` that returns a QoS profile composed of the sum of the throughput parameters (GBRs and MBRs) out of the values obtained for each dynamic PCC Rule running in the IP-CAN session, and selecting the highest value in the rest of the QoS parameters

Table 8 QoS for Bearer Policy Type

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Bearer QoS Control QoS for Bearer	qos	ip-can-session	<subscriberId> <dataplanId>	permit <b>max-qos:</b> BearerQosProfile [ "<qosProfileName>" ] or qos_prof_expression <sup>(1)</sup> permit <b>min-qos:</b>	<b>Mixing policies and qualification</b> Conditions: —Subscriber —Access Data



				BearerQosProfile [ "<qosProfileName>" ] or qos_prof_expression <sup>(1)</sup>	—SubsCharging —Tdf Data ToD
--	--	--	--	---	-----------------------------------

(1) qos\_prof\_expression can be either maxDynQosProfile or sumDynQosProfile.

## 3.12 Configure Access and Charging Policies Based on Dynamic Service Establishment

The SAPC can be configured to reevaluate the access and charging policy decisions taken for the IP-CAN and Sd session based on the successful resource allocation of dynamic services.

The configuration of access and charging policy control is described in *Configuration Guide for Access and Charging Control (Gx)*. In addition, the following policy tag is defined in connection with dynamic services based on notifications received from the Rx and Sd interfaces:

- `AccessData.subscriber.service["serviceName"].isRunning` that indicates if a dynamic service is running in the IP-CAN session. This allows the SAPC to take policy decisions depending on a given dynamic service is running or not.



## 4 Configuration Examples for Use Cases

### 4.1 Upgrade Default Bearer to Prioritize Streaming Delivery

[Example 11](#) shows how application detection can be used to upgrade the default bearer QoS when specific application traffic is detected by the TDF and reported to the SAPC.

The ADC rule configured in the "Netflix" content is activated in the TDF for subscribers belong to "Bronze" subscriber group. When the TDF notifies the SAPC about the start of the application identified as "NetflixStream", the SAPC updates the default bearer QoS in the PCEF.

#### Example 11 Default Bearer QoS Upgrade

```
PUT /contents/Netflix
{
  "contentName" : "Netflix",
  "adcRuleName" : "2000",
  "adcRuleType" : 0
}

PUT /dataplan/Bronze
{
  "dataplanName" : "Bronze",
  "subscribedContents" :
  [
    {
      "contentName" : "Netflix",
      "redirect" : false
    }
  ],
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "MaxBearerQoS",
    "minBearerQosProfileId" : "MinBearerQoS"
  }
}

PUT /profiles/ip-can-session-qos/MinBearerQoS
{
  "arpPci" : true,
  "arpPriorityLevel" : 3,
  "arpPvi" : false,
  "mbrDownlink" : 512,
  "mbrUplink" : 512,
  "profileId" : "MinBearerQoS",
  "qci" : 6
}

PUT /profiles/ip-can-session-qos/MaxBearerQoS
{
  "arpPci" : true,
  "arpPriorityLevel" : 3,
  "arpPvi" : false,
  "mbrDownlink" : 1024,
  "mbrUplink" : 1024,
  "profileId" : "MaxBearerQoS",
  "qci" : 6
}

PUT /profiles/ip-can-session-qos/UpgradeBearerQoS
{
  "arpPci" : true,
  "arpPriorityLevel" : 3,
  "arpPvi" : false,
  "mbrDownlink" : 2048,
```



```
"mbrUplink" : 2048,
"profileId" : "UpgradeBearerQoS",
"qci" : 6
}

PUT /rules/rQosUpgrade
{
  "condition" : "(TdfData.tdfApp[\"NetflixStream\"]').isStarted)",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"UpgradeBearerQoS\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rQosUpgrade"
}

PUT /policies/pQosUpgrade
{
  "policyName" : "pQosUpgrade",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rQosUpgrade" ]
}

PUT /dataplan/Bronze/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pQosUpgrade" ]
}
```

## 4.2 Allocate Dedicated Bearer to Prioritize Streaming Delivery

[Example 12](#) shows how application detection can be used to allocate a dedicated bearer, when specific application traffic is detected by the TDF and reported to the SAPC.

The ADC rule configured in the "YouTube" content is activated in the TDF for subscribers belong to "Premium" subscriber group. When the TDF notifies the SAPC about the starting of the application identified as "YouTubeStream" and delivered from the remote server using ports in the range of 1000-2000, the SAPC requests to initiate a dedicated bearer in the PCEF.

### Example 12 YouTube Dedicated Bearer Allocation"

```
PUT /profiles/content-charging/Char_YouTube
{
  "chargingServiceId" : 301,
  "profileId" : "Char_YouTube",
  "ratingGroup" : 301,
  "reportingLevel" : 1
}

PUT /profiles/content-qos/QoS_YouTube
{
  "arpPriorityLevel" : 7,
  "profileId" : "QoS_YouTube",
  "qci" : 2
}

PUT /contents/YouTube
{
  "contentName" : "YouTube",
  "staticQualification" :
  {
    "contentChargingProfileId" : "Char_YouTube",
```



```

        "contentQosProfileId" : "QoS_YouTube"
    }
}

PUT /rules/rule_YouTube
{
    "condition" : "(TdfData.tdfAppId == \"YouTubeStream\") && inRange(TdfData.port, \"1000-2000\")",
    "outputAttributes" :
    [
        {
            "attrName" : "service",
            "attrValue" : "\"YouTube\"",
            "result" : "permit"
        }
    ],
    "ruleName" : "rule_YouTube"
}

PUT /policies/policy_YouTube
{
    "policyName" : "policy_YouTube",
    "ruleCombiningAlgorithm" : "any-match",
    "rules" : [ "rule_YouTube" ]
}

PUT /locators/resources/tdf-detected-application/contexts/service-classification
{
    "policies" : [ "policy_YouTube" ]
}

```



## 5 Appendix A. ADC over Sd Policy Types

Table 9 and Table 10 show the different policy types applicable to ADC over Sd and Dynamic Policy Control over Sd that can be used and configured in the SAPC.

Table 9 Policy Types for Sd Access and Static Access Qualification

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Access Control (Service Authorization) <b>Access</b>	access	<contentId>	<subscriberId> <dataplanId>		<b>Sd</b> Conditions: —Subscriber —Access Data —SubsCharging —Tdf Data ToD
Access Control (Static Service Qualification) <b>Static Access</b>	static-access	<contentId>	<subscriberId> <dataplanId>	permit <b>adc-rule-id</b> "<adcRuleName>"	<b>Sd</b> Conditions: —Subscriber —Access Data —SubsCharging —Tdf Data ToD

Table 10 Policy Types for Sd Dynamic Service Classification

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
<b>Dynamic Service Classification for Sd</b>	service-classification	tdf-detected-application	-	<b>service</b> <contentId>	Algorithms: —any-match  Conditions: —Tdf Data
Bearer QoS Control (Dynamic Service Qualification) <b>QoS for Service</b>	qos	<contentId>	<subscriberId> <dataplanId>	permit <b>qos</b> ServiceQosProfile ["<qosProfileName>"]	<b>Mixing policies and qualification</b> Conditions: —Subscriber —Access Data —SubsCharging —Tdf Data





(Dynamic Service Qualification) <b>Service Charging</b>	charging	<contentId>	<subscriberId> <dataplanId>	permit <b>charging</b> ServiceChargingProfile ["<chargingProfileName>"]	ToD <b>Mixing policies and qualification</b> Conditions: — Subscriber — Access Data — SubsCharging — Tdf Data ToD
Bearer QoS Control <b>QoS for Bearer</b>	qos	ip-can-session	<subscriberId> <dataplanId>	permit <b>max-qos:</b> BearerQosProfile ["<qosProfileName>"] or qos_prof_expression <sup>(1)</sup> permit <b>min-qos:</b> BearerQosProfile ["<qosProfileName>"] or qos_prof_expression <sup>(1)</sup>	<b>Mixing policies and qualification</b> Conditions: — Subscriber — Access Data — SubsCharging — Tdf Data ToD

(1) qos\_prof\_expression can be either maxDynQosProfile or sumDynQosProfile.



## 6 Appendix B. ADC over Sd Policy Tags

The following Sd policy tags can be used in the condition formula of rules:

Table 11 Sd Policy Tag for Dynamic Service Classification Policies

Tag	Return Type	Possible Values	Comments
<code>TdfData.tdfAppId</code>	String	any	Identifies the TDF application identifier. This tag iterates on each application detection information of the Sd message.
<code>TdfData.tdfAppInstanceId</code>	String	any	Given a TDF application identifier, identifies the instance associated to the flows reported by the TDF. This tag iterates on each application detection information of the Sd message.
<code>TdfData.ipAddress</code>	IPAddress		Identifies the value of the remote server IP. This tag iterates on each application detection information of the Sd message.
<code>TdfData.port</code>	String		Identifies the value of the remote server port. It could also be a port range. This tag iterates on each application detection information of the Sd message.

Table 12 Sd Policy Tag for Qualification Policies

Tag	Return Type	Possible Values	Comments
<code>TdfData.tdfApp["id"].isStarted</code>	Boolean	true/false	<p>The current state for the TDF application:</p> <p><b>True</b></p> <ul style="list-style-type: none"><li>— There are no instances (reporting done at TDF-Application-Id level): when the last event received by the SAPC was a start</li><li>— There are multiple instances: when, for any of the instances under the TDF-Application-Id, the last event received by the SAPC was a start</li></ul> <p><b>False</b></p> <ul style="list-style-type: none"><li>— Before receiving any event from the TDF</li><li>— There are no instances (reporting done at TDF-Application-Id level): when the last application event received by the SAPC was a stop</li><li>— There are multiple instances: when, for every instance under the TDF-Application-Id, the last event received by the SAPC was a stop</li></ul>



## 7 Reference List

### **Ericsson Documents**

1. Configuration Guide for Access and Charging Control (Gx)
2. Configuration Guide for ADC based on PCC rules (Gx)
3. Configuration Guide for Bearer QoS and Bandwidth Management
4. Configuration Guide for Dynamic Policy Control (Rx)
5. Configuration Guide for Subscription and Policies
6. Managed Object Model (MOM)
7. Provisioning REST API