

# Configuration Guide for Fair Usage

Ericsson Service-Aware Policy Controller

USER GUIDE

**Copyright**

© Ericsson España, S.A. 2017. 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.

**Abstract**

This document is a guideline to configure the SAPC for some typical cases related to Fair Usage.



# Contents

<b>1</b>	<b>Configuration and Provisioning Overview</b>	<b>1</b>
1.1	Typographic Conventions	1
1.2	Other Conventions	2
<b>2</b>	<b>Configuration Prerequisites</b>	<b>5</b>
<b>3</b>	<b>Configure Gx PCEFs for Usage Reporting</b>	<b>7</b>
<b>4</b>	<b>Configure Event-Triggers</b>	<b>9</b>
<b>5</b>	<b>Provision Fair Usage Profile</b>	<b>11</b>
5.1	Provision Postpaid Subscriptions	14
5.2	Provision Prepaid Subscriptions	17
5.2.1	At Subscriber Level	18
5.2.2	At Subscriber Group Level	20
5.3	Intermediate Limits	22
5.4	Complementary Limits	23
5.5	Autoprovisioning and Fair Usage	26
<b>6</b>	<b>Provision Shared Subscriber Plans</b>	<b>27</b>
<b>7</b>	<b>Configure Policy Control Based on Fair Usage</b>	<b>31</b>
<b>8</b>	<b>Provision Accumulation Policies</b>	<b>35</b>
8.1	Complementary usage accumulators Based on Conditions	36
<b>9</b>	<b>Usage Reporting Based on Multiple Gx</b>	<b>39</b>
<b>10</b>	<b>Monitor usage accumulators</b>	<b>43</b>
10.1	Monitor Subscriber usage accumulators	43
10.2	Monitor Shared Dataplan usage accumulators	43
<b>11</b>	<b>Configuration Examples for Use Cases</b>	<b>45</b>
11.1	Basic Turbo Button Based on Temporary Subscriber Groups	45
11.2	Advanced Turbo Button with Dynamic Group Selection Based on Fair Usage	49
11.3	Shared Subscriber Plans Use Cases	52
11.3.1	Shared and Individual Limits for Members	52
11.3.2	Category Limits for shared dataplan Members	56
11.3.3	Monitoring Shared Subscriber Plan Members	60



11.3.4	Combining Individual Plans and Shared Subscriber Plan	63
12	<b>Appendix A. Fair Usage Policy Types</b>	<b>71</b>
13	<b>Appendix B. Fair Usage Policy Tags</b>	<b>73</b>
13.1	Fair Usage Policy Tags in Dynamic Group Selection Policies	86
	<b>Glossary</b>	<b>89</b>
	<b>Reference List</b>	<b>91</b>



# 1 Configuration and Provisioning Overview

Next figure, shows the main parts related to configuration and provisioning in the SAPC.

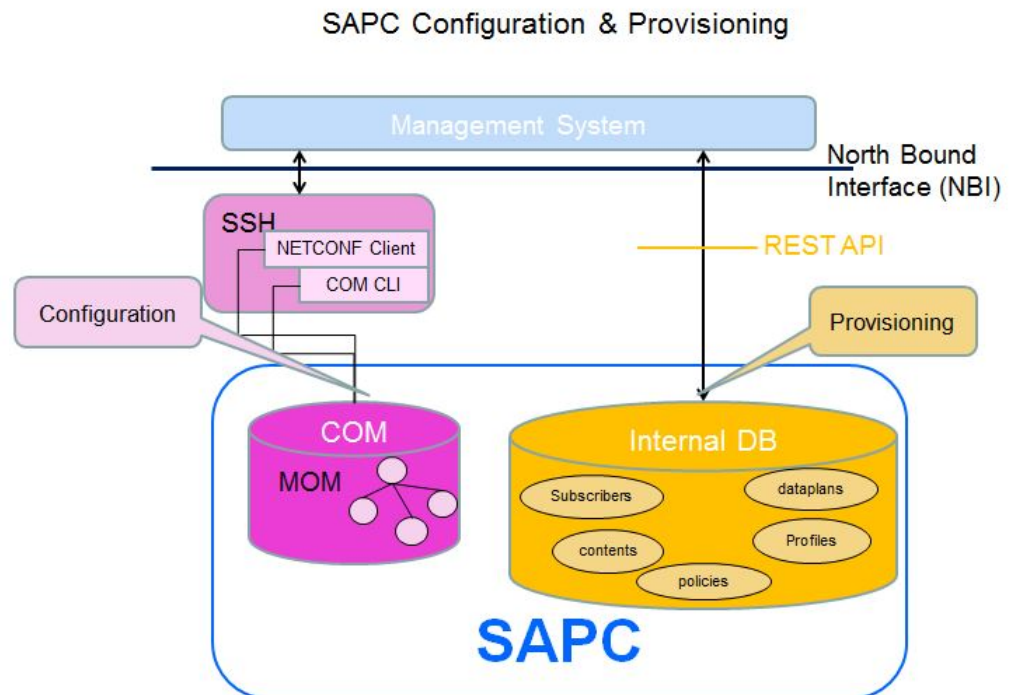


Figure 1 Configuration and Provisioning Overview

The purpose of this document is to provide a guideline to configure Fair Usage in the Ericsson Service-Aware Policy Controller (SAPC) by providing some configuration examples.

This document is not an exhaustive guide for configuring all possibilities related to Fair Usage in the SAPC.

The complete parameter list and details of all configurable options are included in separate documents, refer to [Managed Object Model \(MOM\)](#) and [Provisioning REST API](#).

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

## 1.1 Typographic Conventions

This document uses the following typographic conventions:



Table 1 Typographic Conventions

Convention	Description	Example
MOC	COM Model Object Class	DiameterNode
User Input	A command that you must enter in a Command-Line Interface (CLI) exactly as written.	<pre>PUT /dataplan/GoldSubscriberGroup { "dataplanName" : "GoldSubscriberGroup", "usageLimits" : [ {   "absoluteLimits" : { "dlVolume" : 409600, "resetPeriod" : { "volume" : "monthly" } }, "description" : "Total traffic" } ] }</pre>

**Note:** Within `usageLimits` attribute in REST, the examples along this document include blank spaces only for legibility reasons. To minimize the memory size, **do not** insert any blank spaces.

## 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 for 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:

a 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>
```

b Or 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.

And 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 Configure Gx PCEFs for Usage Reporting

To enable Fair Usage in the SAPC for the PCEF that sends Usage Reporting, set `USAGE_REPORTING` value in `controls` attribute in the corresponding `DiameterNode` instance.





## 4 Configure Event-Triggers

To guarantee that the PCEF reauthorizes (sends a CCR-update to the SAPC) the associated quota when a network event is produced, the SAPC has to include USAGE\_REPORT value within Event-Trigger AVP of Gx CCA-initial message. For details on how to configure event triggers, see [Configuration Guide for Access and Charging Control \(Gx\)](#).





## 5 Provision Fair Usage Profile

To provision a Fair Usage Profile for a subscriber or subscriber group, use `usageLimits` attribute in a subscriber or dataplan URI in the provisioning REST API.

**Note:** For subscriber usage limit, and to prevent issues if Subscriber profile is migrated from monolithic (Subscriber data in the SAPC) to layered solution (subscriber data in UDC), do not exceed as maximum 2680 characters.

The SAPC offers multiple options related to Fair Usage. Page 11 shows the correspondence among Fair Usage options, and the configurable fields within `usageLimits` attribute.

Table 2 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within <code>usageLimits</code>		Comments
Reporting Group	<code>"name": "reportingGroupName"</code>		Identifies the <b>Reporting Group</b> for which Usage Reporting is received in Gx messages.  If a Reporting Group is defined both in the subscriber and subscriber group, the one defined in the subscriber has precedence over the one defined in the group.
Reporting Granularity	<code>"reportingLevel" : "totalTraffic/perReportingGroup"</code>		Usage Limits can apply at total traffic level or Reporting Group level (that is, a service or group of services).
	For <b>total traffic</b>	<code>"name": "reportingGroupName"</code> <code>"reportingLevel": "totalTraffic"</code>	<code>"reportingGroupName"</code> coincides with the usage reported under Monitoring-Key AVP (agreed with the PCEF) within Usage-Monitoring-Information AVP containing Usage-Monitoring-Level AVP set to SESSION_LEVEL.
	<b>Reporting Group</b>	<code>"reportingLevel": "perReportingGroup"</code> <code>"name": "reportingGroupName"</code>	<code>"reportingGroupName"</code> coincides with the usage reported under Monitoring-Key AVP (agreed with the PCEF) within Usage-Monitoring-Information AVP containing Usage-Monitoring-Level AVP set to PCCRULE_LEVEL.



Table 2 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within usageLimits		Comments
Subscription Type	Postpaid See Section 5.1 on page 14	"subscriptionType": "postpaid"	<b>Changes in Subscription Postpaid/Prepaid for a Subscriber:</b>  When the subscription for a Reporting Group changes its type for a subscriber from prepaid to postpaid or the other way around, the accumulated usages are not reset. So, next usage reporting received after the subscription change, are accumulated on the previously stored usage accumulator.
	Prepaid See Section 5.2 on page 17	"subscriptionType": "prepaid"	
Period of usage accumulation	Absolute	"absoluteLimits"	The limits (absolute or complementary) can apply for a period of time or for the IP session duration.
	Complementary  See Section 5.4 on page 22	"conditionalLimits" (1)	
	IP session	"sessionLimits"	





Table 2 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within usageLimits		Comments
Life Cycle of usage accumulators <sup>(2)</sup>	Subscription Date	"subscriptionDate": "dd-mm-yyThh:mm:ss"	<p>When the subscription date is explicitly configured either in the Fair Usage profile for the subscriber or subscriber group, the SAPC considers a subscription date the system date and time at the moment of starting the first session for the subscriber.</p> <p><b>Flexible Billing Cycles:</b> When subscription date value is not provisioned in the Reporting Group of the subscriber, and this Reporting Group corresponds to a temporary dataplan (that is, a subscriber group assigned to the subscriber with an startDate), then subscription date takes its value from the startDate field.</p>
	Reset Period	<pre>"resetPeriod":{   "volume":"monthly"/   "&lt;nr&gt; days"/   "&lt;nr&gt; hours"/   "monthly day &lt;1-31&gt; hh:mm"/   "weekly day &lt;Monday-Sunday&gt; hh:mm"/   "daily hh:mm",   "time": "monthly"/   "&lt;nr&gt; days"/"&lt;nr&gt; hours"/   "monthly day &lt;1-31&gt; hh:mm"/   "weekly day &lt;Monday-Sunday&gt; hh:mm"/   "daily hh:mm" },</pre>	



Table 2 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within usageLimits		Comments
Usage Limit Types	Volume	"ulVolume": [<uplinkVolume>], "dlVolume": [<downlinkVolume>], "bidirVolume": [<bidirectionalVolume>],	
	Time	"time": [<time>]	
Usage Limits split in intervals	"minQuotaTime" : <time>, "minQuotaVolume" : <volume>, "reportingIntervalTime" : <time>, "sliceTime": <time>, "sliceVolume": <volume>,		

- (1) To monitor the complementary usage accumulator (but no need for different policy control actions): set limits to 0 value.  
(2) Considerations with daylight saving time, refer to Configuration Guide for Subscription and Policies

The complete and detailed description for the configurable elements is in Provisioning REST API.

---

---

### Warning!

If a subscriber belongs to several subscriber groups of the same priority, do not provision different Fair Usage Profile values in the different subscriber groups. In that case, the SAPC selects the Fair Usage Profile defined in the first subscriber group.  
For subscriber groups of different priority, there is not conflict, as the SAPC selects the Fair Usage Profile defined in the subscriber group with higher priority.

---

---

## 5.1 Provision Postpaid Subscriptions

Here, an example of provisioning usage limits for total traffic on a postpaid subscriber group:



```
PUT /dataplan/GoldSubscriberGroup
{
  "dataplanName" : "GoldSubscriberGroup",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "dlVolume" : 409600,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      }
    },
    "description" : "Total traffic"
  ]
}
```

#### Example 1 Usage Control for subscriber group with postpaid subscription

For Gold subscribers, downlink traffic volume is limited to 400 Mbytes.

Here, an example of provisioning usage limits for total traffic on a postpaid subscriber:

```
PUT /subscribers/subs0101
{
  "subscriberId" : "subs0101",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : [ 1572864, 2097152 ],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      }
    },
    "description" : "Total traffic"
  ]
}
```

#### Example 2 Usage Control for a subscriber with a postpaid subscription

Next, a new service (PeerToPeer) is provisioned to be controlled based on its usage.



```
PUT /contents/PeerToPeer
{
  "contentName" : "PeerToPeer",
  "flows" :
  [
    {
      "destIpAddr" : "any",
      "destPort" : "",
      "direction" : "dl",
      "flowName" : "1",
      "protocol" : "17",
      "sourceIpAddr" : "153.88.21.22",
      "sourcePort" : "4665"
    },
    {
      "destIpAddr" : "153.88.21.22",
      "destPort" : "4666",
      "direction" : "ul",
      "flowName" : "2",
      "protocol" : "17",
      "sourceIpAddr" : "any",
      "sourcePort" : ""
    }
  ],
  "pccRuleId" : 1234,
  "pccRuleType" : 2
}
```

### Example 3 Provisioning of Peer to Peer service

An example of provisioning usage limits for a postpaid subscriber group is the following:



```

PUT /dataplan/AllInOne
{
  "dataplanName" : "AllInOne",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_AllInOne"
  },
  "subscribedContents" :
  [
    {
      "contentName" : "PeerToPeer",
      "redirect" : false
    }
  ],
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 262144,
        "resetPeriod" :
        {
          "volume" : "7 days"
        },
        "ulVolume" : [ 65536 ]
      },
      "description" : "PeerToPeerFileSharing",
      "name" : "1234"
    },
    {
      "absoluteLimits" :
      {
        "bidirVolume" : [ 1572864, 2097152 ],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "description" : "Total traffic"
    }
  ]
}

PUT /profiles/ip-can-session-qos/QoS_AllInOne
{
  "mbrDownlink" : 1024,
  "mbrUplink" : 1024,
  "profileId" : "QoS_AllInOne",
  "qci" : 5
}

```

#### Example 4 Usage Control for Subscriber Group

For AllInOne subscribers, it is provisioned the following Usage limits:

- For PeerToPeer services (identified by 1234 PCC Rule name), bidirectional volume is limited to 256 Mbytes per week, and uplink to 64 Mbytes per week
- For the whole total traffic, bidirectional volume is limited to 2 Gbytes per month, and there is also an intermediate limit of 1,5 GB.

Moreover, a QoS profile (QoS\_AllInOne) is assigned to the subscriber group.

## 5.2 Provision Prepaid Subscriptions

The SAPC also supports to handle vouchers or promotions, that is, the subscription has an expiry date that is not periodically reset. Instead, subscribers can recharge



their account giving the possibility to extend usage limits and/or expiry date or account refill.

### 5.2.1 At Subscriber Level

Limits for prepaid subscriptions can be provisioned at Subscriber level so that their subscription and expiration date and time is personalized. To provision them, use `usageLimits` attribute in subscriber URI in the provisioning REST API.

---

---

#### Warning!

Do not use prepaid limits s with “unknown” subscriber.

---

---

Next, an example for provisioning prepaid subscription limits, showing a limit of 2 Gbytes for the total traffic for 15 days, starting at 01-09-2020:

```
PUT /subscribers/subs0201
{
  "subscriberId" : "subs0201",
  "trafficIds" : [ "778373000" ],
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
          "volume" : "15 days"
        }
      },
      "description" : "Total traffic",
      "subscriptionDate" : "01-09-2020",
      "subscriptionType" : "prepaid"
    }
  ]
}
```

#### Example 5 Usage Control for Subscriber with prepaid subscription

##### Prepaid Policies

As in the case of postpaid subscriptions, for prepaid subscriptions, apart from provisioning the subscriber limits, it is needed to configure the proper policies to act when the limits are reached or when the subscription has expired.

It is important to use properly `AccessData.subscriber.accumulateUsage.reportingGroup["total"].isActive["<type>"]` and `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].hasExpired["<type>"]` tags within the policy conditions.



## Do!

Delete the provisioning of prepaid Reporting Groups that have already expired, if they are not going to be reused in the future.

## Refill/Recharge

To refill/recharge the subscription, it is needed to modify the subscriber `subscriptionDate` (current accumulated consumed usage are then reset), and if needed, update `usageLimits` (for example if the operator wants to add the remaining unspent quota).

There are several possibilities to do at refill/recharge of the prepaid subscription, each of them requires modification on different elements within `usageLimits` attribute.

Table 3 Refill/Recharge Possibilities

Situation	Action on Subscription Date	Action on Limits	Action on resetPeriod
Restart the subscription (1)	Change to the new subscription date	-	-
Extend the voucher validity (2)	-	-	Change to new voucher validity
Change the limits (3)	-	Change	-
Unspent quota is added to the new limits	-	Monitor usage accumulator (see Section 10 on page 43), and add the difference between the previous limit and the current accumulator.	-

(1) Same conditions apply to the new subscription, that is same limits and same validity

(2) Only possible if subscription has not expired yet

(3) Same validity



Here an example to modify the prepaid subscription for Subscriber **subs0201**, renewing the subscription date to 01-11-2020, changing its validity to 30 days and changing the limit to 2.5 Gbytes. This is the case for example the subscriber had as remaining 0.5 Gbytes unspent, and the operator wants to add them to the previous limit (2 Gbytes).

```
PUT /subscribers/subs0201
{
  "subscriberId" : "subs0201",
  "trafficIds" : [ "778373000" ],
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2621440,
        "resetPeriod" :
        {
          "volume" : "30 days"
        }
      },
      "description" : "Total traffic",
      "subscriptionDate" : "01-11-2020",
      "subscriptionType" : "prepaid"
    }
  ]
}
```

#### Example 6 Prepaid Refill

### 5.2.2 At Subscriber Group Level

Prepaid subscriptions can be also provisioned at subscriber group level and then associate it either to a subscriber or to a shared dataplan (see Section 6 on page 27). Applicability of such prepaid subscription for a subscriber can be controlled using temporary subscription to that group.

The advantage of this approach is that at the end date, the subscription automatically (without OAM intervention) takes the characteristics of the normal group.





```

PUT /dataplan/PrepaidGoldSubscriberGroup
{
  "dataplanName" : "PrepaidGoldSubscriberGroup",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
          "volume" : "7 days"
        }
      },
      "description" : "Total traffic",
      "subscriptionType" : "prepaid"
    }
  ]
}

PUT /subscribers/subs0201
{
  "dataplan" :
  [
    {
      "dataplanName" : "PrepaidGoldSubscriberGroup",
      "priority" : 1,
      "startDate" : "01-07-2020",
      "stopDate" : "31-07-2020"
    }
  ],
  "subscriberId" : "subs0201",
  "trafficIds" : [ "778373000" ],
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2621440,
        "resetPeriod" :
        {
          "volume" : "30 days"
        }
      },
      "description" : "Total traffic",
      "subscriptionDate" : "01-11-2020",
      "subscriptionType" : "prepaid"
    }
  ]
}

```

### Example 7 Provisioning of Prepaid Subscriber Group

Previous example shows how 2 Gbytes of volume limit and a validity of 7 days is provisioned for a Prepaid subscriber group.

Subscriber subs0201 is assigned temporary to the Prepaid Gold Group using as startDate, endDate the dates corresponding to the period where the subscriber has the possibility of buying the prepaid voucher (01-07-2020 until 31-07-2020).

### Refill/Recharge

In this case, the refill for the voucher can also be done by modifying the attribute startDate (within subscriber URI in the provisioning REST API. . When the new start date is reached, the accumulated consumed usage are reset.



```
PUT /subscribers/subs0201
{
  "dataplan" :
  [
    {
      "dataplanName" : "PrepaidGoldSubscriberGroup",
      "priority" : 1,
      "startDate" : "01-09-2020",
      "stopDate" : "30-09-2020"
    }
  ],
  "subscriberId" : "subs0201",
  "trafficIds" : [ "778373000" ]
}
```

Example 8 Refill at subscriber group level

## 5.3 Intermediate Limits

To set intermediate limits for a Reporting Group, use an array in `ulVolume`, `dlVolume`, `bidirVolume` or time attributes.

**Note:** When **intermediate** limits are provisioned for a Reporting Group and they are used within policy conditions, use array notation: the first limit (starting from left to right) is specified by using `[0]`, the second one as `[1]` and so on. Do configure them from lower to higher values (to ease the tracking of threshold values reached).

Do follow the same ordering in the array element used in the conditions than in the limits.

In Example 2, `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].isLimitSurpassed["bidirVolume"][0]` points to 1.5 Gbytes and `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].isLimitSurpassed["bidirVolume"][1]` points to 2 Gbytes for “subs0101” subscriber.

If the array contains only one limit element, the notation `[0]` is not needed.

In Example 1, `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].isLimitSurpassed["dlVolume"]` points to 400 Mbytes.

If the array contains several limit elements, `isLimitSurpassed["<type>"]` points to the first of them (that is, it is equivalent to `isLimitSurpassed["<type>"][0]`).

In Example 2, `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].isLimitSurpassed["bidirVolume"]` points to 1.5 Gbytes.



## 5.4 Complementary Limits

To set different limits for different reset periods in a Reporting Group, use `ulVolume`, `dlVolume`, `bidirVolume` or `time` and `resetPeriod` block attributes inside `conditionalLimits` block.

**Note:** In this case, it is not needed to configure Accumulation conditional policies (Section 8 on page 35).

Next figure shows the absolute and complementary limits:



Figure 2 Complementary and Absolute Limits

If `resetPeriod` is not defined inside `conditionalLimits`, the SAPC takes the reset period value of the absolute limit.

The following example shows a way to set for a Reporting Group different usage limits for different reset periods. It can be interesting to set a volume limit (4 Gbytes) that is reset monthly, but also other Complementary limits: a daily limit of 500 MB and a weekly limit of 2 Gbytes. Once the subscriber reaches a limit, the QoS is downgraded.

Next, a graphical summary of the configuration:

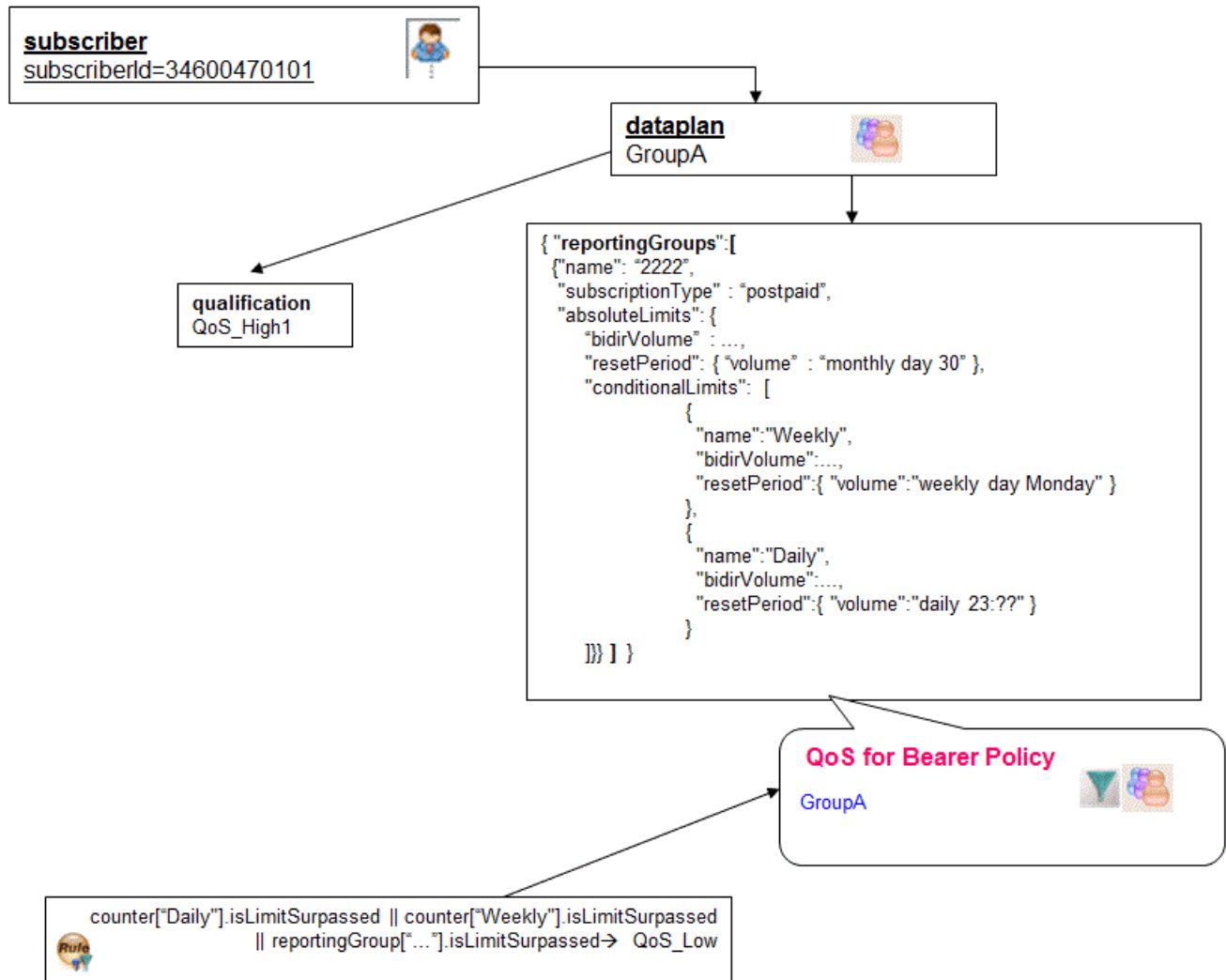


Figure 3 Complementary Limits with different reset periods

```
PUT /rules/rQoS_Low1
{
  "condition" : "(((AccessData.subscriber.accumulatedUsage.reportingGroup[\"2222\"]).
    counter[\"Daily\"]).isLimitSurpassed[\"bidirVolume\"])) ||
    ((AccessData.subscriber.accumulatedUsage.reportingGroup[\"2222\"]).
    counter[\"Weekly\"]).isLimitSurpassed[\"bidirVolume\"])) ||
    ((AccessData.subscriber.accumulatedUsage.reportingGroup[\"2222\"]).
    isLimitSurpassed[\"bidirVolume\"])) ",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low1\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low1\"]",
      "result" : "permit"
    }
  ]
},
```



```

    "ruleName" : "rQoS_Low1"
  }
PUT /policies/pQoS_Low1
{
  "policyName" : "pQoS_Low1",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rQoS_Low1" ]
}
PUT /dataplans/GroupA/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pQoS_Low1" ]
}
PUT /dataplans/GroupA
{
  "dataplanName" : "GroupA",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_High1"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 4194304,
        "conditionalLimits" :
        [
          {
            "bidirVolume" : 2097152,
            "name" : "Weekly",
            "resetPeriod" :
            {
              "volume" : "weekly day Monday"
            }
          },
          {
            "bidirVolume" : 512000,
            "name" : "Daily",
            "resetPeriod" :
            {
              "volume" : "daily 23:??"
            }
          }
        ]
      },
      "resetPeriod" :
      {
        "volume" : "monthly day 30"
      },
      "name" : "2222",
      "sliceVolume" : 256000,
      "subscriptionType" : "postpaid"
    }
  ]
}
PUT /profiles/ip-can-session-qos/QoS_High1
{
  "mbrDownlink" : 6144,
  "mbrUplink" : 6144,
  "profileId" : "QoS_High1",
  "qci" : 5
}
PUT /profiles/ip-can-session-qos/QoS_Low1
{
  "mbrDownlink" : 1024,
  "mbrUplink" : 1024,
  "profileId" : "QoS_Low1",
  "qci" : 5
}
PUT /subscribers/34600470101

```



```
{
  "dataplan" :
  [
    {
      "dataplanName" : "GroupA"
    }
  ],
  "subscriberId" : "34600470101"
}
```

#### Example 9 Configuration for different complementary limit/reset periods

In the example above, the subscriber belongs to “GroupA”. The reset period for the Reporting Group is defined as “monthly day 30”. There are also complementary limits: “Daily” and “Weekly”. Each of them with the corresponding reset period. Two QoS profiles are defined, so when the subscriber's usage surpasses the daily, weekly or absolute limits, the bearer QoS is downgraded to “QoS\_Low1”. The “Daily” usage accumulator is reset every day at “23:??”, meaning any time during that hour, while “Weekly” is reset every Monday. In case the QoS would have been downgraded to “QoS\_Low1” due to surpassing the “Daily” limit, once the complementary usage accumulator is reset, the QoS profile applied is “QoS\_High1” again.

## 5.5 Autoprovisioning and Fair Usage

For cases where Autoprovisioned and Fair Usage are combined: do only use “subscriptionType” = “postpaid” in the usageLimits of the “auto” dataplan.

Once the subscriber is automatically provisioned, and usage reporting is received for the subscriber, the content of accumulatedUsage contains the usage accumulator.



## 6 Provision Shared Subscriber Plans

The SAPC allows to define Shared Subscriber Plans, where several subscribers report usage to the same usage accumulator. The Fair Usage Profiles associated to that accumulator are shared by all the members associated to the plan. The subscribers typically might be part of a domestic family or an enterprise. An individual subscriber cannot belong to several shared plans simultaneously.

Next figure shows the relations of the shared dataplan related data:

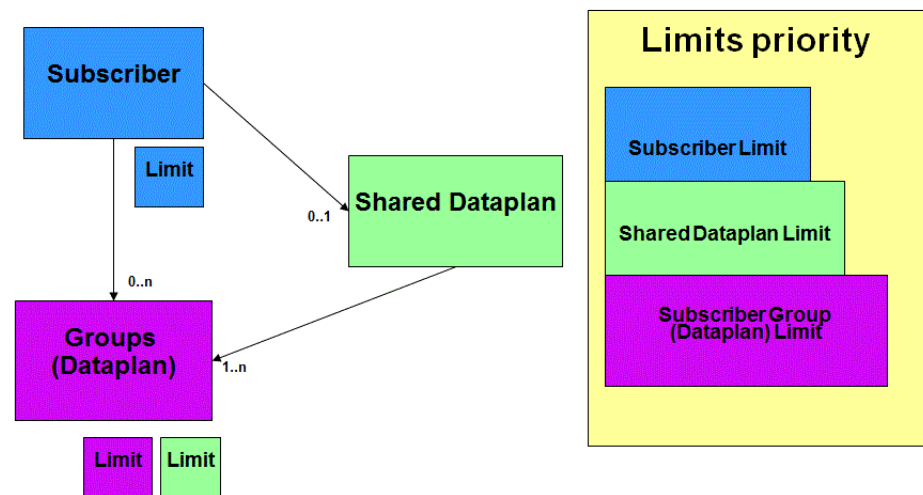


Figure 4 Shared Dataplan Relations

The relations shown above correspond to the default behavior. The Fair Usage Profile defined for the subscriber has priority over the one defined for the shared dataplan if both contain the same Reporting Group. Furthermore, if the subscriber belongs to any other group different from the shared dataplan, and such group contains a Fair Usage Profile for the same Reporting Group, the Fair Usage Profile for the shared dataplan with higher priority prevails over the rest of subscriber groups.

The SAPC allows to alter the precedence between shared dataplan and Subscriber Groups using Dynamic Group Selection policies for the subscriber. For further information, see Section 11.3 on page 52.

The usage limits that are shared among the members of the shared dataplan are the **absolute limits**, including both general and conditional limits.

To provision Fair Usage Profile for shared dataplans:

1. Create the shared dataplan, using `/shared-dataplans/{sharedDataplanId}` URI in the provisioning REST API



2. Set the Fair Usage Profile for the shared dataplan, using `usageLimits` attribute for the corresponding dataplan URI in the provisioning REST API
3. Set `sharedDataplan` attribute for the subscribers belonging to the shared dataplan

**Subscription Date for groups:** To have different subscription dates for two shared dataplans using the same Service Offering (same value of `dataplanName` attribute), set different `startDate` (within `dataplans` attribute) of each shared dataplan.

**Note:** It is not possible to use the `subscriptionDate` attribute inside the Reporting Group.

---



---

## Warning!

Do not use following element for shared dataplan (as the SAPC does not support it):

Session accumulation.

---



---

Next, an example is shown:

```
PUT /rules/QoS_NormalBW
{
  "condition" : "not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
                 isLimitSurpassed[\"bidirVolume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Normal\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Normal\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "QoS_NormalBW"
}

PUT /rules/QoS_ReducedBW
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
                 isLimitSurpassed[\"bidirVolume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Reduced\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Reduced\"]",
      "result" : "permit"
    }
  ],
}
```





```

    "ruleName" : "QoS_ReducedBW"
  }

  PUT /policies/pBandwidthLimit
  {
    "policyName" : "pBandwidthLimit",
    "ruleCombiningAlgorithm" : "permit-overrides",
    "rules" : [ "QoS_ReducedBW", "QoS_NormalBW" ]
  }

  PUT /dataplan/BasicFamilyOffering/locators/resources/ip-can-session/contexts/qos
  {
    "policies" : [ "pBandwidthLimit" ]
  }

  PUT /dataplan/BasicFamilyOffering
  {
    "dataplanName" : "BasicFamilyOffering",
    "usageLimits" :
    [
      {
        "absoluteLimits" :
        {
          "bidirVolume" : 5242880,
          "resetPeriod" :
          {
            "volume" : "monthly"
          }
        },
        "description" : "Total traffic",
        "sliceVolume" : 50
      }
    ]
  }

  PUT /shared-dataplan/Garcia
  {
    "dataplan" :
    [
      {
        "dataplanName" : "BasicFamilyOffering"
      }
    ],
    "sharedDataplanId" : "Garcia"
  }

  PUT /profiles/ip-can-session-qos/QoS_Normal
  {
    "mbrDownlink" : 512,
    "mbrUplink" : 512,
    "profileId" : "QoS_Normal",
    "qci" : 5
  }

  PUT /profiles/ip-can-session-qos/QoS_Reduced
  {
    "mbrDownlink" : 128,
    "mbrUplink" : 128,
    "profileId" : "QoS_Reduced",
    "qci" : 5
  }

  PUT /subscribers/34610101010
  {
    "sharedDataplan" : "Garcia",
    "subscriberId" : "34610101010"
  }

  PUT /subscribers/34620202020
  {
    "sharedDataplan" : "Garcia",
    "subscriberId" : "34620202020"
  }

```

#### Example 10 Shared Subscriber Plan for a Family



This example provisions subscribers “34610101010” and “34620202020” belonging to “Garcia” family. Garcia family uses the offering (and Fair Usage Profile) provisioned by BasicFamilyOffering (typically, other family instances, for example Castro, Lopez and Martin could reuse the same offering). 50 Gbytes are provided as limit for the BasicFamilyOffering, using slices of 50 KB. When the limit is surpassed, the bandwidth is reduced to QoS\_Reduced.



## 7

## Configure Policy Control Based on Fair Usage

When the subscriber reaches the provisioned limits, the SAPC can deny the access to the service, change the bandwidth, apply different ratings, and so on, depending on the provisioned policies. This section shows how to configure the SAPC to act when usage limits are surpassed.

Next, a typical example for a condition based on Usage Reporting limits to authorize or not a Service:

```
PUT /rules/Auth_VoIP
{
  "condition" : "not (AccessData.subscriber.session.accumulatedUsage.reportingGroup[\"5001\"]).isLimitSurpassed[\"time\"]",
  "ruleName" : "Auth_VoIP"
}

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

PUT /dataplan/AllInOne/locators/resources/Skype/contexts/access
{
  "policies" : [ "Auth_VoIP" ]
}

PUT /dataplan/AllInOne
{
  "dataplanName" : "AllInOne"
}
```

### Example 11 Configuration for Service Authorization based on Usage Control

A policy for AllInOne subscribers is configured so that Skype service is only authorized if the time limit per session is not exceeded.

It is also possible to authorize or not a service using usage limits of a different service: for example Internet service could be authorized only if total volume traffic is not exceeded.

Here an example of how configuring a policy to assign bearer QoS profile to the subscriber depending on Fair Usage.



```

PUT /rules/RestrictQoS
{
  "condition" : "AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
                 isLimitSurpassed[\"bidirVolume\"]",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Restricted\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Restricted\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "RestrictQoS"
}

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

PUT /dataplans/AllInOne/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "RestrictQoS" ]
}

PUT /profiles/content-qos/QoS_Restricted
{
  "mbrDownlink" : 128,
  "mbrUplink" : 128,
  "profileId" : "QoS_Restricted",
  "qci" : 5
}

```

### Example 12 Bearer QoS based on Usage Control

A policy for AllInOne subscribers is configured, so that a QoS Profile "QoS\_Restricted" (the QoS values for this profile are lower than the ones for the QoS profile statically assigned for AllInOne group) is assigned to be downloaded to PCEF when the total volume traffic is exceeded (see Example 4).

Next an example showing next characteristics for subscriber group "MobileBroadband":

- When the consumed total traffic does not exceed the limit and the prepaid subscription is active, internet service is qualified to be charged using flat rate, using CR-Name = 2001.
- When the consumed total traffic surpasses 2 Gbytes and the prepaid subscription is active, internet service is qualified to be throttled (bandwidth limited) but still flat rate applies, using CR-Name = 2002.
- When the prepaid subscription expires, internet service is qualified to be charged with no flat rate, using CR-Name = 2003.



```

PUT /rules/rFlatRate
{
  "condition" : "not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
    isLimitSurpassed[\"bidirVolume\"])&&
    (AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].isActive[\"volume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "pcc-rule-id",
      "attrValue" : "2001",
      "result" : "permit"
    }
  ],
  "ruleName" : "rFlatRate"
}

PUT /rules/rNotFlatRate
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].hasExpired[\"
  "outputAttributes" :
  [
    {
      "attrName" : "pcc-rule-id",
      "attrValue" : "2003",
      "result" : "permit"
    }
  ],
  "ruleName" : "rNotFlatRate"
}

PUT /rules/rThrottling
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
    isLimitSurpassed[\"bidirVolume\"])&&
    (AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].isActive[\"volume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "pcc-rule-id",
      "attrValue" : "2002",
      "result" : "permit"
    }
  ],
  "ruleName" : "rThrottling"
}

PUT /policies/pQualifyPrepaid
{
  "policyName" : "pQualifyPrepaid",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rFlatRate", "rThrottling", "rNotFlatRate" ]
}

PUT /dataplan/MobileBroadband/locators/resources/Internet/contexts/static-access
{
  "policies" : [ "pQualifyPrepaid" ]
}

```

### Example 13 Policies for Prepaid





## 8 Provision Accumulation Policies

The operator can decide under what conditions the accumulation of usage is done, for example accumulate only when the subscriber is connected during rush hours.

This can be done for all the Reporting Groups defined in the SAPC (using reporting-group) or specifically for a specific Reporting Group (<reportingGroupName>).

To configure conditional usage accumulation, create the needed policies using:

— For **Global policy locator**:

```
/locators/resources/reporting-group/contexts/accumulation or
/locators/resources/<reportingGroupName>/contexts/accumulation
```

— For **Subscriber group locator**:

```
/dataplanes/<dataplanName>/locators/resources/reporting-group/co
ntexts/accumulation or /dataplanes/<dataplanName>/locators/resour
ces/<reportingGroupName>/contexts/accumulation
```

— For **Subscriber locator**:

```
/subscribers/<subscriberId>/locators/resources/reporting-group/
contexts/accumulation or /subscribers/<subscriberId>/locators/re
sources/<reportingGroupName>/contexts/accumulation
```

For the conditional reporting to work properly, it is also required that the SAPC receives a request including the usage reporting when the condition changes.

Next an example for a global policy (applicable for all subscribers) that accumulates from 8:00 to 18:00.

```
PUT /rules/ruleAccumRushHours
{
  "condition" : "((now.time>\\"08:00\\")&&(now.time<\\"18:00\\"))",
  "ruleName" : "ruleAccumRushHours"
}

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

PUT /locators/resources/total/contexts/accumulation
{
  "policies" : [ "acumPolicy" ]
}
```

### Example 14 Policies for Accumulation



## 8.1 Complementary usage accumulators Based on Conditions

The usage can be accumulated in different usage accumulator configured for each of the conditions.

For multiple usage accumulators, configure following things:

- 1 Within `usageLimits` attribute, use the `conditionalLimits` block to specify the usage accumulator names and their limits
- 2 To specify the conditions that dictate the accumulation on the particular counters, create policies using:

- a For usage accumulator applicable to all Reporting Groups:

```
<policy locator URI>/resources/reporting-group.<counterName>/contexts/accumulation
```

- b or for usage accumulator of a particular Reporting Group:

```
<policy locator URI>/resources/reportingGroupName.<counterName>/contexts/accumulation
```

where `<policy locator URI>` can be on of the following:

- **Global policy locator:** `/locators/`
- **Subscriber group locator:** `/dataplan/<dataplanName>/locators/`
- **Subscriber locator:** `/subscribers/<subscriberId>/locators/`

---

### Warning!

The usage accumulator name has to be the same within `conditionalLimit` block and the resource of the policies.

---

The policies applicable for `<reportingGroupName>.<counterName>` have precedence over the ones defined for `reporting-group.<counterName>`. That is, conditions for `reporting-group.<counterName>` applies to all counters of all the Reporting Groups that have not particular accumulation conditions, including the total traffic.

Next an example to accumulate usage depending on ToD:

```
PUT /rules/rAcumFlatRateHours
{
  "condition" : "((now.time>\\"00:00\\")&&(now.time<\\"08:00\\"))||((now.time>\\"17:00\\")&&(now.time<\\"23:59\\"))",
  "ruleName" : "rAcumFlatRateHours"
}

PUT /rules/rAcumNotFlatRateHours
```





```

{
  "condition" : "((now.time>\\"08:00\\")&&(now.time<\\"17:00\\"))",
  "ruleName" : "rAcumNotFlatRateHours"
}

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

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

PUT /dataplan/Premium/locators/resources/total.FlatRateHours/contexts/accumulation
{
  "policies" : [ "pAccumFlatRateHours" ]
}

PUT /dataplan/Premium/locators/resources/total.NotFlatRateHours/contexts/accumulation
{
  "policies" : [ "pAccumNotFlatRateHours" ]
}

PUT /dataplan/Premium
{
  "dataplanName" : "Premium",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "conditionalLimits" :
        [
          {
            "dlVolume" : 1024,
            "name" : "NotFlatRateHours"
          },
          {
            "dlVolume" : 2048,
            "name" : "FlatRateHours"
          }
        ],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "description" : "Total traffic",
      "subscriptionDate" : "25-12-2020T08:00:00"
    }
  ]
}

PUT /subscribers/subs0208
{
  "dataplan" :
  [
    {
      "dataplanName" : "Premium",
      "priority" : 1
    }
  ],
  "subscriberId" : "subs0208",
  "trafficIds" : [ "mary@ericsson.com" ]
}

```

## Example 15 Conditional Accumulation





## 9 Usage Reporting Based on Multiple Gx

The typical scenario for Multiple Gx is when several PCEF's control the same IP-CAN-Session. One PCEF supports Usage Reporting. The other PCEF does not support Usage Reporting but performs the control enforcement (for example Bearer QoS Control). The PCEF supporting Usage Reporting sends the consumed usage to the SAPC. The SAPC, when detects that usage limits are surpassed, sends a reauthorization to the other PCEF.

Refer to [Configuration Guide for Access and Charging Control \(Gx\)](#) for other configuration details about Multiple Gx.

---

---

### Warning!

Only subscriber absolute limits per period and use of `AccessData.subscriber.accumulatedUsage.reportingGroup["reportingGroupName"].isLimitSurpassed["type"]` in the condition formula) have sense in this Multiple Gx scenario.

Do not use Session limits in this scenario.

---

---

Next example assumes a case where a subscriber belonging to AllInOne group establishes several sessions:



```

PUT /rules/rule_QoS_throttled
{
  "condition" : "AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"].
                isLimitSurpassed[\"bidirVolume\"]",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"Qos_Throttled\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rule_QoS_throttled"
}

PUT /rules/rule_Qos1
{
  "condition" : "1",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QosProfile1\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rule_Qos1"
}

PUT /policies/pQoS_throttled
{
  "policyName" : "pQoS_throttled",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rule_QoS_throttled", "rule_Qos1" ]
}

PUT /dataplan/AllInOne/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "RestrictQoS", "pQoS_throttled" ]
}

PUT /profiles/ip-can-session-qos/QosProfile1
{
  "mbrDownlink" : 1024,
  "mbrUplink" : 128,
  "profileId" : "QosProfile1",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/Qos_Throttled
{
  "mbrDownlink" : 128,
  "mbrUplink" : 128,
  "profileId" : "Qos_Throttled",
  "qci" : 5
}

```

#### Example 16 Usage Control based on multiple Gx

- Gx requests received from a PCEF supporting Bearer QoS Control (for example a GGSN) where a QoS Profile of 1024 Kbps (for the downlink) is applied as result of Bearer QoS Control policies (ruleId= rule\_Qos1).
- Gx requests received from a PCEF (for example a SASN) and acting as usage reporter (Usage Reporting control activated), are mainly used to accumulate the consumed usages in the SAPC

When the total traffic bidirectional consumed usage reaches the limit (provisioned in Example 4), the QoS Profile assigned towards the GGSN shall be changed: a RAR message is to be sent including this QoS change



data to 128 Kbps (for the downlink). This can be achieved by the use of `ruleId=rule_QoS_throttled`.

Be aware that the most restrictive condition for the rules (in this case the condition about the surpassed limit) inside the same policy has to be configured first, as permit overrides combination algorithm is implicitly used.

Configuration regarding other controls (such as IP session Access Control, Service Access Control, Content Filtering ) also applicable to these Gx requests is not explicitly included in this example, but can be found in [Configuration Guide for Access and Charging Control \(Gx\)](#).





## 10 Monitor usage accumulators

### 10.1 Monitor Subscriber usage accumulators

To read the content of a subscriber usage accumulators, use GET operation for `/subscribers/{subscriberId}/usage-accumulators` URI in the provisioning REST API.

### 10.2 Monitor Shared Dataplan usage accumulators

To read the content of a shared dataplan usage accumulators, use GET operation for `/shared-dataplan/{sharedDataplanId}/usage-accumulators` URI in the provisioning REST API.







# 11 Configuration Examples for Use Cases

## 11.1 Basic Turbo Button Based on Temporary Subscriber Groups

Consider a subscriber with a normal postpaid Fair Usage subscription (for example 2 Gbytes at 1 Mbps, and when the limit is reached, the speed is throttled to 64 Kbps), that orders during a certain period of time a higher bandwidth with different limits (for example 250 MB at 3 Mbps, from 20:00 to 22:00 of the 1st of January).

This is like ordering a high-speed voucher with time limitation. It is possible to model this use case defining two subscriber groups. One called normal, containing the basic subscription, and another one called turbo, including the turbo-button characteristics.

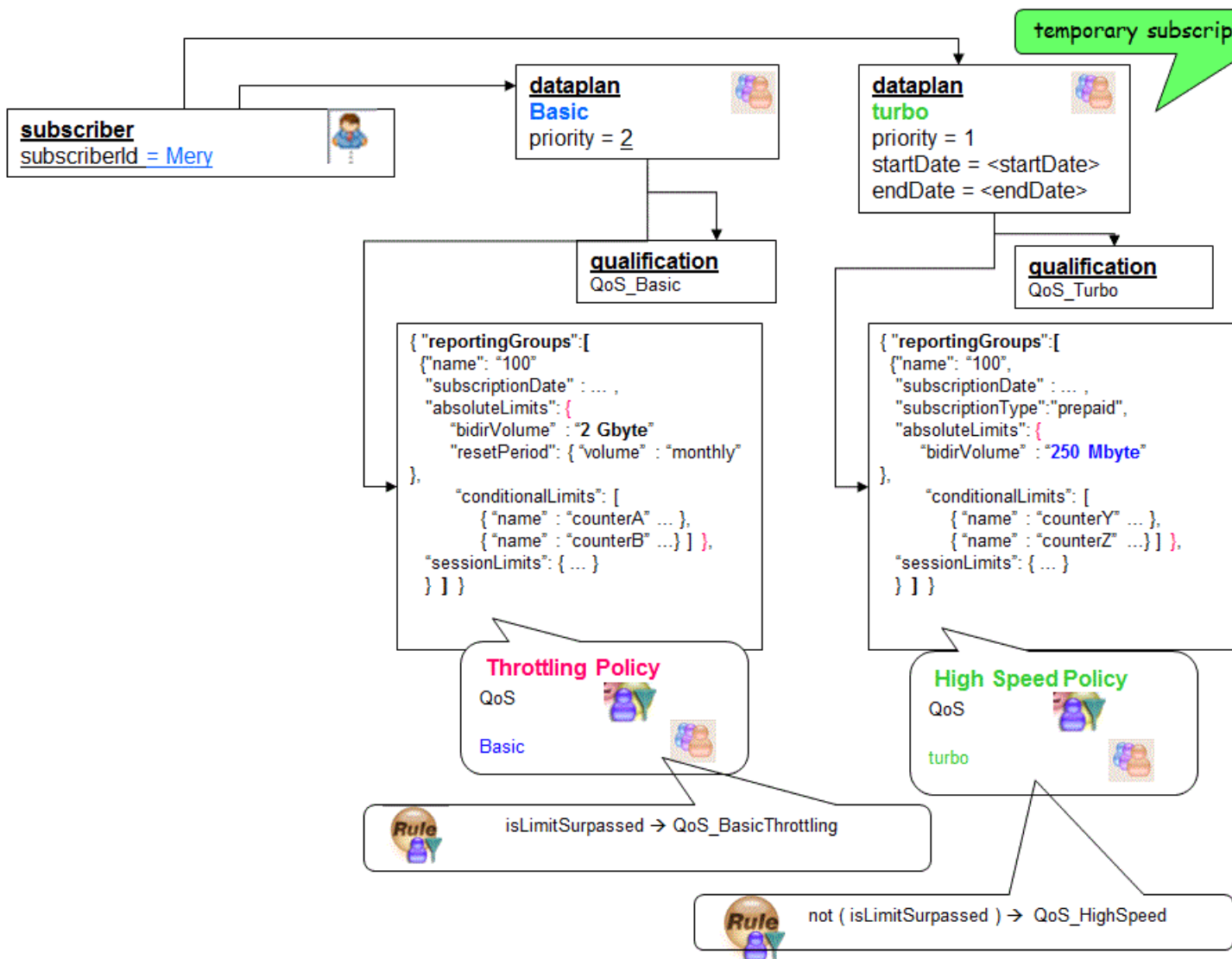


Figure 5 Basic Turbo Button based on Temporary Subscriber Groups

```
PUT /rules/rHighSpeed3M
{
  "condition": "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"100\"].group[\"Turbo\"].remaining[\"bidirVolume\"] > 1",
  "outputAttributes": [
    {
      "attrName": "max-qos",
      "attrValue": "BearerQosProfile[\"QoS_HighSpeed\"]",
      "result": "permit"
    },
    {
      "attrName": "min-qos",
      "attrValue": "BearerQosProfile[\"QoS_HighSpeed\"]",
      "result": "permit"
    }
  ],
  "ruleName": "rHighSpeed3M"
}

PUT /rules/rThrottling64K
```



```
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"100\"].
    group[\"Basic\"].remaining[\"bidirVolume\"]) < 1",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_BasicThrottling\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_BasicThrottling\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rThrottling64K"
}

PUT /policies/pHighSpeed3M
{
  "policyName" : "pHighSpeed3M",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rHighSpeed3M" ]
}

PUT /policies/pThrottling64K
{
  "policyName" : "pThrottling64K",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rThrottling64K" ]
}

PUT /dataplans/Basic/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pThrottling64K" ]
}

PUT /dataplans/Turbo/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pHighSpeed3M" ]
}

PUT /dataplans/Basic
{
  "dataplanName" : "Basic",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_Basic"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "description" : "Peer to Peer",
      "name" : "100"
    }
  ]
}

PUT /dataplans/Turbo
{
  "dataplanName" : "Turbo",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_Turbo"
  },
  "usageLimits" :
  [
```



```

    {
      "absoluteLimits" :
      {
        "bidirVolume" : 256000
      },
      "description" : "Peer to Peer",
      "name" : "100",
      "subscriptionType" : "prepaid"
    }
  ]
}

PUT /profiles/ip-can-session-qos/QoS_Basic
{
  "mbrDownlink" : 1024,
  "mbrUplink" : 1024,
  "profileId" : "QoS_Basic",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS_BasicThrottling
{
  "mbrDownlink" : 64,
  "mbrUplink" : 64,
  "profileId" : "QoS_BasicThrottling",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS_HighSpeed
{
  "mbrDownlink" : 3072,
  "mbrUplink" : 3072,
  "profileId" : "QoS_HighSpeed",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS_Turbo
{
  "mbrDownlink" : 2048,
  "mbrUplink" : 2048,
  "profileId" : "QoS_Turbo",
  "qci" : 5
}

PUT /subscribers/34600002411
{
  "dataplan" :
  [
    {
      "dataplanName" : "Basic",
      "priority" : 2
    },
    {
      "dataplanName" : "Turbo",
      "priority" : 1,
      "startDate" : "01-01-2002T20",
      "stopDate" : "01-01-2002T22"
    }
  ],
  "subscriberId" : "34600002411"
}

```

### Example 17 Basic Turbo Button based on Temporary Subscriber Groups

In the example above, two different subscriptions are provisioned:

- Basic subscribers, associated to Basic group having a statically assigned Bearer QoS profile "QoS\_Basic" and 2 Gbytes volume limit per month
- Turbo subscribers, associated to Turbo group which also has a statically assigned Bearer QoS profile "QoS\_Turbo", and 250 Mbytes volume limit.



Subscriber with traffic identifier 34600002411 belongs to both Basic and Turbo groups, having Turbo group higher priority and temporary applicability from 20:00 to 22:00 of the 1st of January.

Depending on the volume limits being surpassed or not, a different Bearer QoS profile is dynamically (by policies) assigned:

- For Basic subscribers, when the limit is surpassed, the bandwidth is throttled by downloading the Bearer QoS profile "QoS\_BasicThrottling" (so QoS is downgraded to 64 Kbps)
- During the Turbo activation, and meanwhile the limit is not surpassed, the bandwidth is enhanced by downloading the Bearer QoS profile "QoS\_HighSpeed" (meaning QoS is set to 3 Mbps). Once the limit of 250 MB is surpassed the bandwidth is not speed up, so the Bearer QoS profile applied is "QoS\_Turbo".

## 11.2 Advanced Turbo Button with Dynamic Group Selection Based on Fair Usage

Consider a subscriber with a normal postpaid Fair Usage subscription (for example 800 Mbytes at 1 Mbps), that orders a plan with higher priority during a certain period having higher bandwidth with different limits (for example 400 Mbytes at 6 Mbps, from January the 1st). To model this, a Dynamic Group Selection policy is defined so the prepaid plan is only active while it is not expired and there is available quota.

Next a graphical sketch of the needed configuration:

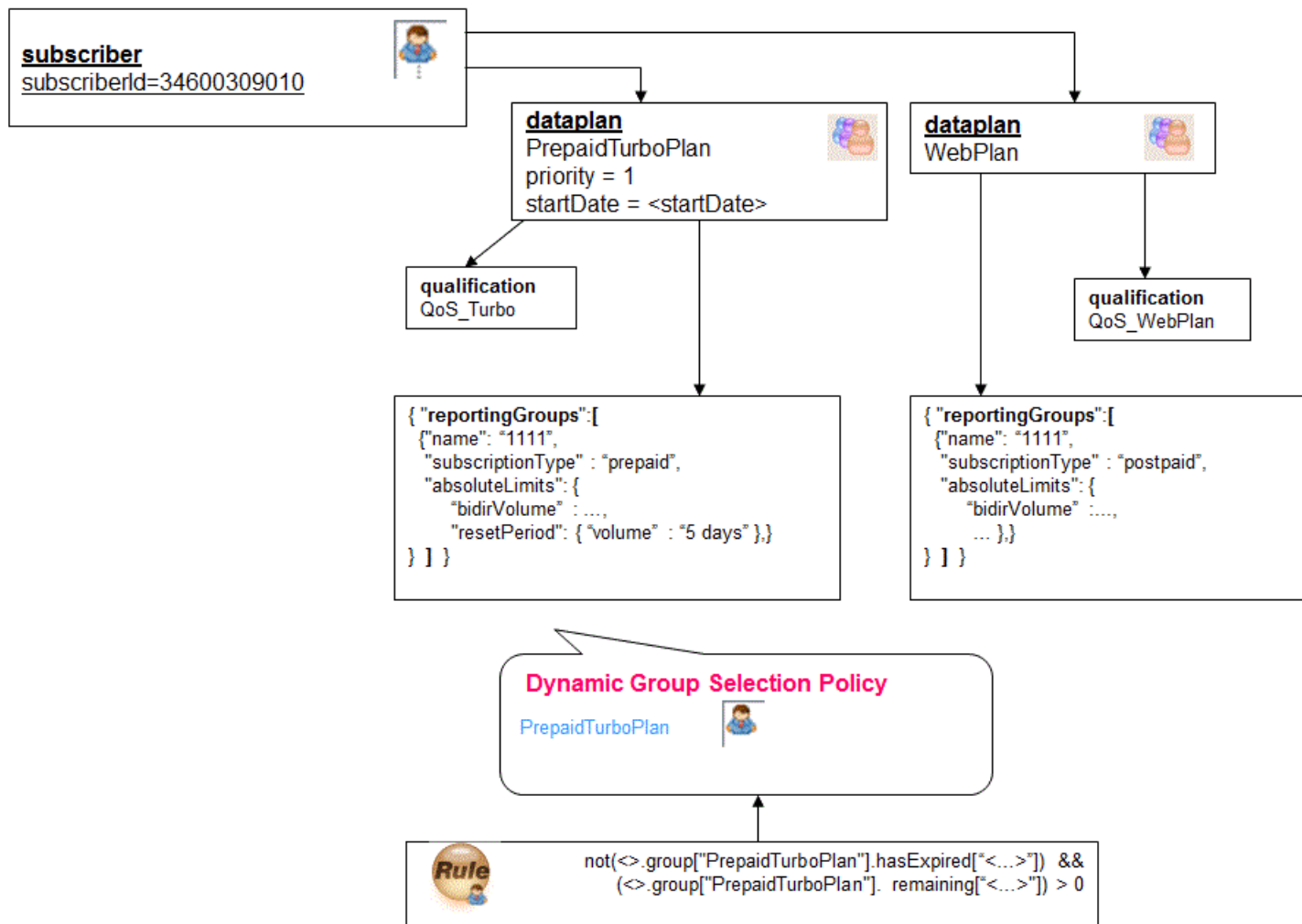


Figure 6 Advanced Turbo Button with Dynamic Group Selection based on Fair Usage

```

PUT /rules/rPrepaid
{
  "condition": "not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\\\"].
    group[\"PrepaidTurboPlan\\\"].hasExpired[\"volume\\\"]) &&
    ((AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\\\"].
    group[\"PrepaidTurboPlan\\\"].remaining[\"bidirVolume\\\"]) > 0)",
  "ruleName": "rPrepaid"
}

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

PUT /locators/resources/PrepaidTurboPlan/contexts/subscription
{
  "policies": [ "pPrepaid" ]
}

PUT /dataplan/PrepaidTurboPlan

```



```

{
  "dataplanName" : "PrepaidTurboPlan",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_Turbo"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 409600,
        "resetPeriod" :
        {
          "volume" : "5 days"
        }
      },
      "name" : "1111",
      "subscriptionType" : "prepaid"
    }
  ]
}

PUT /dataplan/WebPlan
{
  "dataplanName" : "WebPlan",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_WebPlan"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : [ 102400, 512000, 819200 ],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "1111",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /profiles/ip-can-session-qos/QoS_Turbo1
{
  "mbrDownlink" : 6144,
  "mbrUplink" : 6144,
  "profileId" : "QoS_Turbo1",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS_WebPlan
{
  "mbrDownlink" : 1024,
  "mbrUplink" : 1024,
  "profileId" : "QoS_WebPlan",
  "qci" : 5
}

PUT /subscribers/34600309010
{
  "dataplan" :
  [
    {
      "dataplanName" : "WebPlan"
    },
    {
      "dataplanName" : "PrepaidTurboPlan",
      "priority" : 1,
      "startDate" : "01-01-2020"
    }
  ],

```



```
"subscriberId" : "34600309010"  
}
```

#### Example 18 Advanced Turbo Button with Dynamic Group Selection based on Fair Usage

In the example above, the subscriber is associated with “WebPlan” and “PrepaidTurboPlan”. As set in the dataplans, “PrepaidTurboPlan” has a higher priority than the postpaid one and as start date “01-01-2020”. As the reset period has been defined with “5 days” as value, the “PrepaidTurboPlan” expires the “06-01-2020”. The default priority specified in dataplans is altered through Dynamic Group Selection policies. In the condition, there are two conditions to control whether the “PrepaidTurboPlan” is applied or not. The first one is used to check that the “PrepaidTurboPlan” is only active from “01-01-2020” to “01-06-2020”. The second condition uses remaining tag specifying the group to control that there is quota left for “PrepaidTurboPlan”. If any of these conditions is not fulfilled, the service offering applied is “WebPlan”.

For further information regarding the usage of tags for fair usage policies, see Section 13.1 on page 86

## 11.3 Shared Subscriber Plans Use Cases

### 11.3.1 Shared and Individual Limits for Members

Consider a shared dataplan (for example 1.2 Gbytes to be shared among them). Subscriber 1, the one who pays, decides to limit the usage of the other members. To model this, it is necessary to define a conditional limit for those members and accumulation policies so the reporting group becomes disabled for accumulation when the corresponding limit is reached.

Next a figure summarizing the configuration:



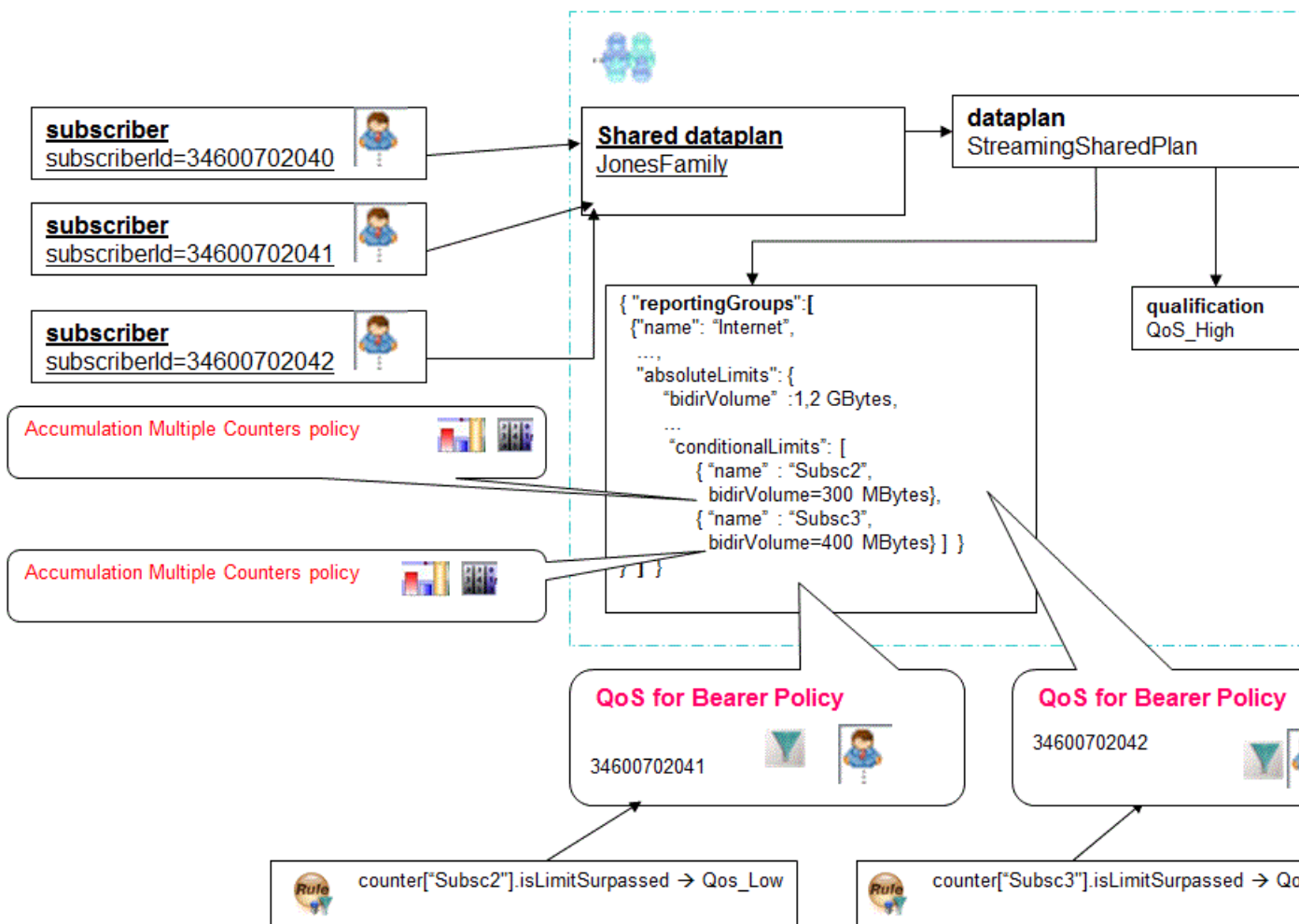


Figure 7 Shared and Individual Limits for Members

```

PUT /rules/RSubsc2
{
  "condition" : "(AccessData.subscriber.id==\"34600702041\")",
  "ruleName" : "RSubsc2"
}

PUT /rules/RSubsc2_RG_Accum
{
  "condition" : "not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
    counter[\"Subsc2\"].isLimitSurpassed[\"bidirVolume\"])",
  "ruleName" : "RSubsc2_RG_Accum"
}

PUT /rules/RSubsc3
{
  "condition" : "(AccessData.subscriber.id==\"34600702042\")",
  "ruleName" : "RSubsc3"
}

PUT /rules/RSubsc3_RG_Accum
{
  "condition" : "not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
    counter[\"Subsc3\"].isLimitSurpassed[\"bidirVolume\"])",

```



```
"ruleName" : "RSubsc3_RG_Accum"
}

PUT /rules/rQoS_Low_Subsc2
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
                  counter[\"Subsc2\"].isLimitSurpassed[\"bidirVolume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rQoS_Low_Subsc2"
}

PUT /rules/rQoS_Low_Subsc3
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
                  counter[\"Subsc3\"].isLimitSurpassed[\"bidirVolume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\"QoS_Low\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rQoS_Low_Subsc3"
}

PUT /policies/PSubsc2
{
  "policyName" : "PSubsc2",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "RSubsc2" ]
}

PUT /policies/PSubsc2_RG_Accum
{
  "policyName" : "PSubsc2_RG_Accum",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "RSubsc2_RG_Accum" ]
}

PUT /policies/PSubsc3
{
  "policyName" : "PSubsc3",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "RSubsc3" ]
}

PUT /policies/PSubsc3_RG_Accum
{
  "policyName" : "PSubsc3_RG_Accum",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "RSubsc3_RG_Accum" ]
}

PUT /policies/pQoS_Subsc2
{
  "policyName" : "pQoS_Subsc2",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rQoS_Low_Subsc2" ]
}
```



```

}
PUT /policies/pQoS_Subsc3
{
  "policyName" : "pQoS_Subsc3",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rQoS_Low_Subsc3" ]
}

PUT /dataplan/StreamingSharedPlan/locators/resources/1111.Subsc2/contexts/accumulation
{
  "policies" : [ "PSubsc2" ]
}

PUT /dataplan/StreamingSharedPlan/locators/resources/1111.Subsc3/contexts/accumulation
{
  "policies" : [ "PSubsc3" ]
}

PUT /subscribers/34600702041/locators/resources/1111/contexts/accumulation
{
  "policies" : [ "PSubsc2_RG_Accum" ]
}

PUT /subscribers/34600702042/locators/resources/1111/contexts/accumulation
{
  "policies" : [ "PSubsc3_RG_Accum" ]
}

PUT /subscribers/34600702041/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pQoS_Subsc2" ]
}

PUT /subscribers/34600702042/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pQoS_Subsc3" ]
}

PUT /dataplan/StreamingSharedPlan
{
  "dataplanName" : "StreamingSharedPlan",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_High"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 1228800,
        "conditionalLimits" :
        [
          {
            "bidirVolume" : 307200,
            "name" : "Subsc2"
          },
          {
            "bidirVolume" : 409600,
            "name" : "Subsc3"
          }
        ]
      },
      "resetPeriod" :
      {
        "volume" : "monthly"
      }
    },
    {
      "name" : "1111"
    }
  ]
}

PUT /shared-dataplan/JonesFamily
{
  "dataplan" :

```



```
[
  {
    "dataplanName" : "StreamingSharedPlan"
  },
  "sharedDataplanId" : "JonesFamily"
]

PUT /profiles/ip-can-session-qos/QoS_High
{
  "mbrDownlink" : 5000,
  "mbrUplink" : 5000,
  "profileId" : "QoS_High",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS_Low
{
  "mbrDownlink" : 1800,
  "mbrUplink" : 1800,
  "profileId" : "QoS_Low",
  "qci" : 5
}

PUT /subscribers/34600702040
{
  "sharedDataplan" : "JonesFamily",
  "subscriberId" : "34600702040"
}

PUT /subscribers/34600702041
{
  "sharedDataplan" : "JonesFamily",
  "subscriberId" : "34600702041"
}

PUT /subscribers/34600702042
{
  "sharedDataplan" : "JonesFamily",
  "subscriberId" : "34600702042"
}
```

#### Example 19 Conditional limits for members of a Shared Subscriber Plan

In this example, the member “34600702041” and “34600702042” of “JonesFamily”, have a usage accumulator, associated to each one of them where the usage report performed by them is accumulated. For the subscriber 2, that limit is 300 Mbytes, while for subscriber 3 is 400 Mbytes. The corresponding usage accumulator is deactivated once the subscriber surpasses the limit due to accumulation policies, and the QoS profile is downgraded to QoS\_Low.

### 11.3.2 Category Limits for shared dataplan Members

This section shows a use case where it is needed to classify the members of a Shared Subscriber Plan and apply different usage limits depending on the member's category. To do so, use the `operatorSpecificInfos` attribute of the subscriber to store the member's type. Then, depending on the category, different policies are applied, in this case, the QoS is downgraded once the limit is surpassed. The “head” category includes subscriber A and it represents the subscriber who pays the Shared Subscriber Plan. The “regular” category includes the other members and represents the subscribers who benefits from the Shared Subscriber Plan.



Next a graphical sketch of the configuration:

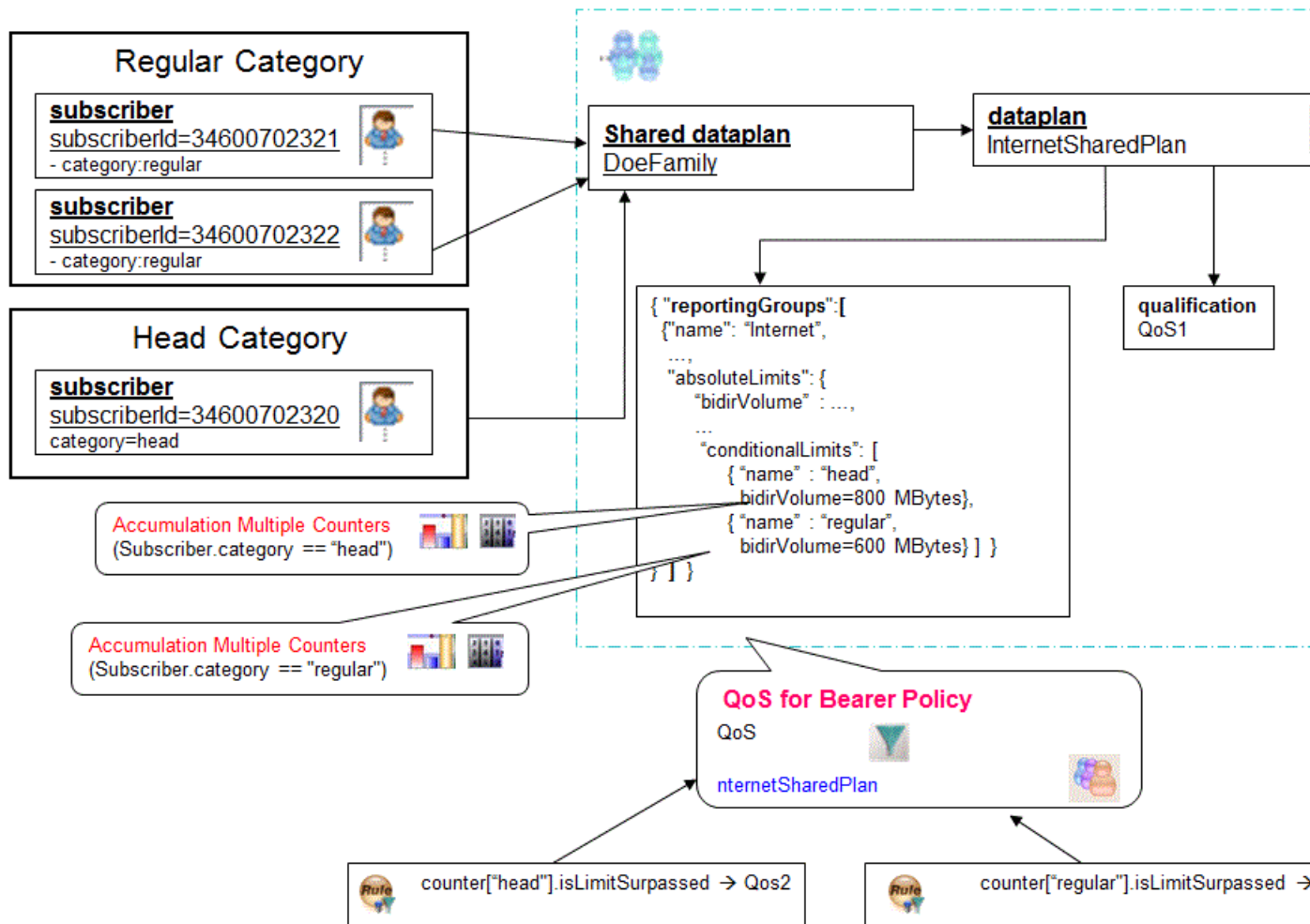


Figure 8 Category Limits for shared dataplan Members

```

PUT /rules/RHead
{
  "condition" : "(Subscriber.category == \"head\")",
  "ruleName" : "RHead"
}

PUT /rules/RRegular
{
  "condition" : "(Subscriber.category == \"regular\")",
  "ruleName" : "RRegular"
}

PUT /rules/rQoS2_head
{
  "condition" : "(Subscriber.category==\"head\") &&
    (AccessData.subscriber.accumulatedUsage.reportingGroup[\"Internet\"].
      counter[\"head\"].isLimitSurpassed[\"bidirVolume\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\"QoS2\"]",
    }
  ]
}

```



```
        "result" : "permit"
      },
      {
        "attrName" : "min-qos",
        "attrValue" : "BearerQosProfile[\\"QoS2\\"]",
        "result" : "permit"
      }
    ],
    "ruleName" : "rQoS2_head"
  }
}

PUT /rules/rQoS2_regular
{
  "condition" : "(Subscriber.category==\\"regular\\") &&
    (AccessData.subscriber.accumulatedUsage.reportingGroup[\\"Internet\\"].
      counter[\\"regular\\"].isLimitSurpassed[\\"bidirVolume\\"])",
  "outputAttributes" :
  [
    {
      "attrName" : "max-qos",
      "attrValue" : "BearerQosProfile[\\"QoS2\\"]",
      "result" : "permit"
    },
    {
      "attrName" : "min-qos",
      "attrValue" : "BearerQosProfile[\\"QoS2\\"]",
      "result" : "permit"
    }
  ],
  "ruleName" : "rQoS2_regular"
}

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

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

PUT /policies/pThrottlingQoS
{
  "policyName" : "pThrottlingQoS",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "rQoS2_head", "rQoS2_regular" ]
}

PUT /dataplans/InternetSharedPlan/locators/resources/Internet.head/contexts/accumulation
{
  "policies" : [ "PHead" ]
}

PUT /dataplans/InternetSharedPlan/locators/resources/Internet.regular/contexts/accumulation
{
  "policies" : [ "PRegular" ]
}

PUT /dataplans/InternetSharedPlan/locators/resources/ip-can-session/contexts/qos
{
  "policies" : [ "pThrottlingQoS" ]
}

PUT /dataplans/InternetSharedPlan
{
  "dataplanName" : "InternetSharedPlan",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS1"
  },
  "usageLimits" :
```



```
[
  {
    "absoluteLimits" :
    {
      "bidirVolume" : 1228800,
      "conditionalLimits" :
      [
        {
          "bidirVolume" : 819200,
          "name" : "head"
        },
        {
          "bidirVolume" : 614400,
          "name" : "regular"
        }
      ],
      "resetPeriod" :
      {
        "volume" : "monthly"
      }
    },
    "name" : "Internet",
    "sliceVolume" : 20480
  }
]
}

PUT /shared-dataplans/DoeFamily
{
  "dataplans" :
  [
    {
      "dataplanName" : "InternetSharedPlan"
    }
  ],
  "sharedDataplanId" : "DoeFamily"
}

PUT /profiles/ip-can-session-qos/QoS1
{
  "mbrDownlink" : 3000,
  "mbrUplink" : 3000,
  "profileId" : "QoS1",
  "qci" : 5
}

PUT /profiles/ip-can-session-qos/QoS2
{
  "mbrDownlink" : 512,
  "mbrUplink" : 512,
  "profileId" : "QoS2",
  "qci" : 5
}

PUT /subscribers/34600702320
{
  "operatorSpecificInfos" :
  [
    {
      "attributeName" : "category",
      "attributeValue" : "head"
    }
  ],
  "sharedDataplan" : "DoeFamily",
  "subscriberId" : "34600702320"
}

PUT /subscribers/34600702321
{
  "operatorSpecificInfos" :
  [
    {
      "attributeName" : "category",
      "attributeValue" : "regular"
    }
  ]
}
```



```

    }
  ],
  "sharedDataplan" : "DoeFamily",
  "subscriberId" : "34600702321"
}

PUT /subscribers/34600702322
{
  "operatorSpecificInfos" :
  [
    {
      "attributeName" : "category",
      "attributeValue" : "regular"
    }
  ],
  "sharedDataplan" : "DoeFamily",
  "subscriberId" : "34600702322"
}

```

### Example 20 Conditional counters based on the category of the member

For this example, the members of “DoeFamily” are divided in two categories. Subscribers “34600702321” and “34600702322” are defined under “regular” category, while “34600702320” is defined as “head”. The reports from subscribers “34600702021” and “34600702322” are accumulated in the “regular” category usage accumulator as well as in the general one. There are also two QoS profiles defined. QoS1 has been defined statically as the subscriber qualification for “InternetSharedPlan”. There are QoS for bearer policies defined so the QoS is downgraded to QoS2 in case the members of any category surpass the corresponding limit.

The conditional formula makes reference to the category of the subscriber and the specific limit: `(Subscriber.category=="regular") && (<>.counter["regular"].isLimitSurpassed["bidirVolume"])`. Otherwise, if the first condition was not present, once a member of the “regular” category surpassed the limit, the members of other categories would have been affected.

## 11.3.3 Monitoring Shared Subscriber Plan Members

An interesting case is when there is the need to monitor the usage performed by each subscriber or category of subscribers, but there is no need of taking a policy control action when a limit is reached. This can be achieved by configuring multiple usage accumulator depending on conditions with a limit of 0, meaning no limit is applied, so only the shared limit is controlled.

When it is desired to monitor subscribers, the Reporting Group has to be defined as in the following example:

```

PUT /rules/RSubscA
{
  "condition" : "(AccessData.subscriber.id=="34600702020")",
  "ruleName" : "RSubscA"
}

PUT /rules/RSubscB
{
  "condition" : "(AccessData.subscriber.id=="34600702021")",
  "ruleName" : "RSubscB"
}

```





```

PUT /rules/RSubscC
{
  "condition" : "(AccessData.subscriber.id==\"34600702022\")",
  "ruleName" : "RSubscC"
}

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

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

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

PUT /dataplan/GroupSharedPlan/locators/resources/1111.SubscA/contexts/accumulation
{
  "policies" : [ "PSubscA" ]
}

PUT /dataplan/GroupSharedPlan/locators/resources/1111.SubscB/contexts/accumulation
{
  "policies" : [ "PSubscB" ]
}

PUT /dataplan/GroupSharedPlan/locators/resources/1111.SubscC/contexts/accumulation
{
  "policies" : [ "PSubscC" ]
}

PUT /dataplan/GroupSharedPlan
{
  "dataplanName" : "GroupSharedPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 1228800,
        "conditionalLimits" :
        [
          {
            "bidirVolume" : 0,
            "name" : "SubscA"
          },
          {
            "bidirVolume" : 0,
            "name" : "SubscB"
          },
          {
            "bidirVolume" : 0,
            "name" : "SubscC"
          }
        ],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "1111",
      "sliceVolume" : 10240
    }
  ]
}

```



```

}
PUT /shared-dataplans/SmithFamily
{
  "dataplanName" :
  [
    {
      "dataplanName" : "GroupSharedPlan"
    }
  ],
  "sharedDataplanId" : "SmithFamily"
}

PUT /subscribers/34600702020
{
  "sharedDataplan" : "SmithFamily",
  "subscriberId" : "34600702020"
}

PUT /subscribers/34600702021
{
  "sharedDataplan" : "SmithFamily",
  "subscriberId" : "34600702021"
}

PUT /subscribers/34600702022
{
  "sharedDataplan" : "SmithFamily",
  "subscriberId" : "34600702022"
}

```

#### Example 21 Configuration to monitor members of a shared data plan

To monitor a specific category, the same principle is applied, but in this case the conditional limit is defined for the category:

```

PUT /dataplan/FamilyDataPlan
{
  "dataplanName" : "FamilyDataPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 1228800,
        "conditionalLimits" :
        [
          {
            "bidirVolume" : 0,
            "name" : "head"
          },
          {
            "bidirVolume" : 0,
            "name" : "regular"
          }
        ]
      },
      "resetPeriod" :
      {
        "volume" : "monthly"
      }
    },
    {
      "name" : "1111",
      "sliceVolume" : 20480
    }
  ]
}

```

#### Example 22 Configuration to monitor categories of members



## **11.3.4 Combining Individual Plans and Shared Subscriber Plan**

The following examples show combinations of Shared Subscriber Plan with individual plans. The examples use a similar configuration, so the full configuration is described in the first one while in the others, to simplify, are presented the differences. To be able to apply Dynamic Group Selection Policies, the individual plan is defined as a subscriber group.

### **11.3.4.1 Individual and Shared Plans for Different Reporting Groups**

An Individual Plan is required for one of the subscribers. When the individual limit is surpassed, it is wanted to apply a Shared Subscriber Plan. In this example, the Individual and the Shared Subscriber Plan use different Reporting Groups and the subscriber that has both groups, starts to report usage to the Individual plan until its limit is surpassed, moment in which the Shared Subscriber Plan is applicable.

Next a graphical sketch of the configuration:

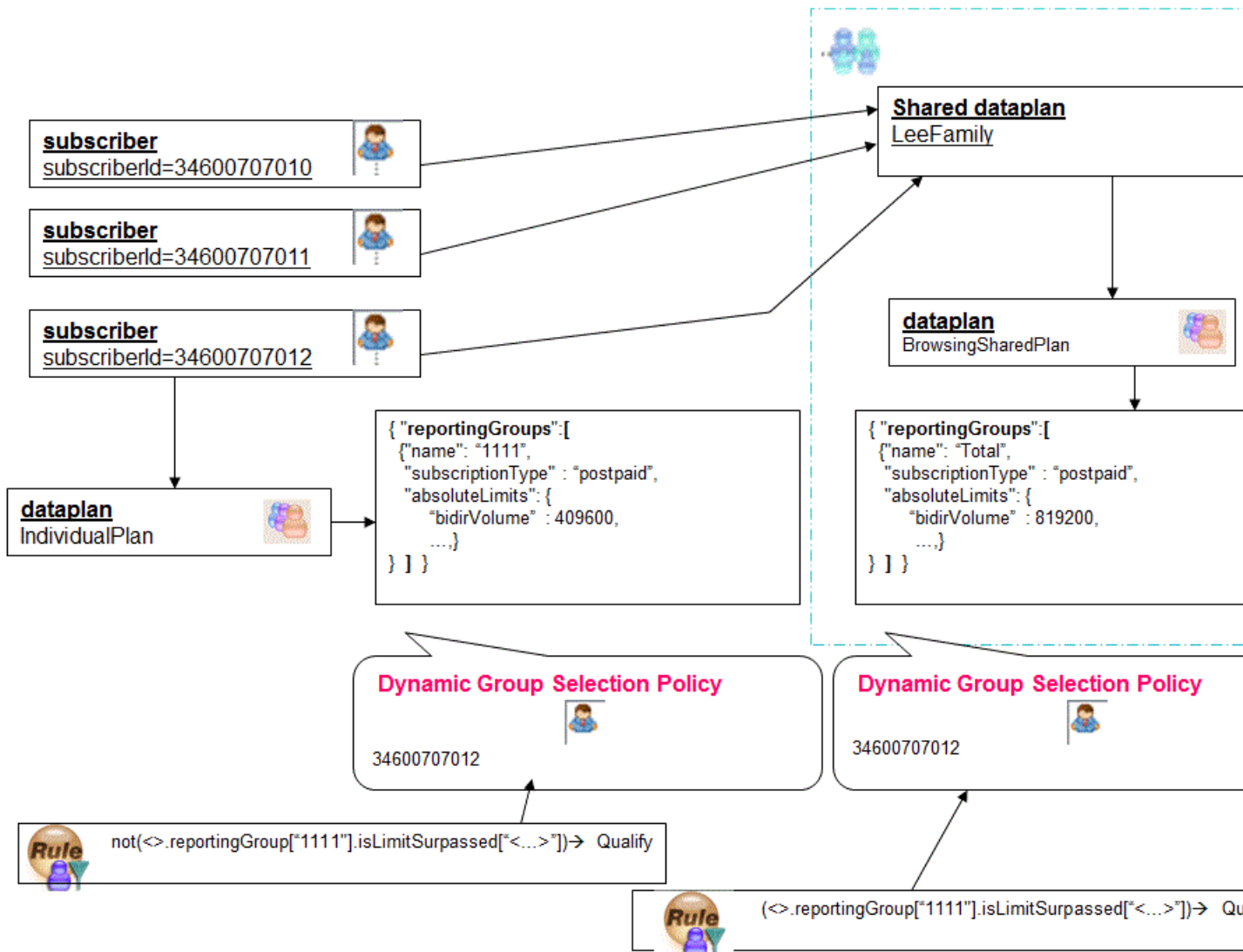


Figure 9 Individual and Shared Plans for Different Reporting Groups

```

PUT /rules/rBrowsingShared
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
    isLimitSurpassed[\"bidirVolume\"])",
  "ruleName" : "rBrowsingShared"
}

PUT /rules/rIndividualPlan
{
  "condition" : "(not(AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"].
    isLimitSurpassed[\"bidirVolume\"]))",
  "ruleName" : "rIndividualPlan"
}

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

```



```

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

PUT /subscribers/34600707012/locators/resources/BrowsingSharedPlan/contexts/subscription
{
  "policies" : [ "pBrowsingShared" ]
}

PUT /subscribers/34600707012/locators/resources/IndividualPlan/contexts/subscription
{
  "policies" : [ "pIndividualPlan" ]
}

PUT /dataplan/BrowsingSharedPlan
{
  "dataplanName" : "BrowsingSharedPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 819200,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "Total",
      "reportingLevel" : "totalTraffic",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /dataplan/IndividualPlan
{
  "dataplanName" : "IndividualPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 409600,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "1111",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /shared-dataplan/LeeFamily
{
  "dataplan" :
  [
    {
      "dataplanName" : "BrowsingSharedPlan"
    }
  ],
  "sharedDataplanId" : "LeeFamily"
}

PUT /subscribers/34600707010
{
  "sharedDataplan" : "LeeFamily",
  "subscriberId" : "34600707010"
}

PUT /subscribers/34600707011

```



```
{
  "sharedDataplan" : "LeeFamily",
  "subscriberId" : "34600707011"
}

PUT /subscribers/34600707012
{
  "dataplan" :
  [
    {
      "dataplanName" : "IndividualPlan"
    }
  ],
  "sharedDataplan" : "LeeFamily",
  "subscriberId" : "34600707012"
}
```

#### Example 23 Combination of Individual and Shared Subscriber Plans

In the example above, the members of “LeeFamily”, subscribers 1, 2 and 3, have “Total” Reporting Group defined. Subscriber “34600707012” has also an individual plan associated. As the two Reporting Groups have different names, the Dynamic Group Selection policies are based on the Individual Reporting Group: `<>.reportingGroup["1111"].isLimitSurpassed["bidirVolume"]`. For further information regarding the usage of tags for fair usage policies, see Section 13.1 on page 86

#### 11.3.4.2

#### Plans for the Same Reporting Group Prioritizing the Individual Plan

It is wanted to apply a particular Individual plan for a subscriber. When the Individual limit is surpassed, it is wanted to apply a Shared Subscriber Plan. There is a conflict between the two Reporting Groups as both of them have the same name. Because of default precedences, the Shared Subscriber Plan is selected, so to alter the precedences, Dynamic Group Selection policies are used.



```

PUT /rules/r3GSharedPlan
{
  "condition" : "((AccessData.subscriber.accumulatedUsage.reportingGroup[\"3333\"].
    group[\"PersonalPlan\"].remaining[\"bidirVolume\"]) == 0)",
  "ruleName" : "r3GSharedPlan"
}

PUT /policies/p3GSharedPlan
{
  "policyName" : "p3GSharedPlan",
  "ruleCombiningAlgorithm" : "permit-overrides",
  "rules" : [ "r3GSharedPlan" ]
}

PUT /subscribers/34600807012/locators/resources/3GSharedPlan/contexts/subscription
{
  "policies" : [ "p3GSharedPlan" ]
}

PUT /dataplan/3GSharedPlan
{
  "dataplanName" : "3GSharedPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 1024000,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "3333",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /dataplan/PersonalPlan
{
  "dataplanName" : "PersonalPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 512000,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "3333",
      "subscriptionType" : "postpaid"
    }
  ]
}

```

#### Example 24 Combination of Individual and Shared Data plan, prioritizing Individual Plan

In the configuration above, it is described the definition of the Reporting Groups and the rules for the Dynamic Group Selection policies. In this case, as the Reporting Groups have the same name “3333”, the conditional uses the policy tag remaining, that allows group granularity:  $((<>.group["PersonalPlan"].remaining["bidirVolume"]) > 0)$ . For further information regarding policy tags for fair usage policies, see Section 13.1 on page 86



### 11.3.4.3 Plans for the Same Reporting Group Prioritizing the Shared Subscriber Plan

In this example, it is the Shared Subscriber Plan that it is consumed first. Heavy users have another data plan and reserve it until the shared one is consumed (for example: an executive that needs connectivity 24/7 but also shares the contract with his/her family).

```
PUT /rules/rShareInternetPlan
{
  "condition" : "((AccessData.subscriber.accumulatedUsage.reportingGroup[\"1111\"]).group[\"ShareInternetPlan\"].current[\"bidirVolume\"] < 1024000)",
  "ruleName" : "rShareInternetPlan"
}

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

PUT /subscribers/34600907012/locators/resources/ShareInternetPlan/contexts/subscription
{
  "policies" : [ "pShareInternetPlan" ]
}

PUT /dataplan/IndividualDataPlan
{
  "dataplanName" : "IndividualDataPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 512000,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "1111",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /dataplan/ShareInternetPlan
{
  "dataplanName" : "ShareInternetPlan",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 1024000,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "name" : "1111",
      "subscriptionType" : "postpaid"
    }
  ]
}
```

Example 25 Combination of Individual and Shared Data plan, prioritizing Shared Data Plan





The configuration above shows the definition of the Reporting Groups and the rules for the Dynamic Group Selection policies. As the Reporting Groups have the same name “1111”, the condition uses the policy tag `current`, that allows group granularity: `((<>.group["ShareInternetPlan"].current["bidirVolume"]) > 1024000)`. For further information regarding policy tags for fair usage policies, see Section 13.1 on page 86.





# 12      Appendix A. Fair Usage Policy Types

Next figure shows the different policy types related to Fair Usage.







Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
Conditional Accumulation	accumulation 	reporting-group <reporting-group-id> 	<subscriberId> <dataplanId> 	-	Type I = Only policies, no qualification Used to accumulate or not depending on flexible policy conditions
Accumulation Multiple Counters	accumulation 	reporting-group .<counterId> <reporting-group-id>.<counterId> 	<subscriberId> <dataplanId> 	-	Type I = Only policies, no qualification Used to accumulate on different counters

Figure 10    Fair Usage Policy Types In the SAPC





## 13 Appendix B. Fair Usage Policy Tags

The following tags, related to Fair Usage may be used in the policy conditions. Owing to the nature of Fair Usage, these tags can be used in any of the SAPC policy types for which this functionality has sense, for example (and typically) Service Access Control, Bearer QoS Control or End-User Notifications.

**Note:** For legibility reasons, policy tags are shown in the following tables without escaping double quotes (") character, but it is needed to use backslash (\) before them.

Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Value of the Reporting Group usage accumulator selected for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. expiryDate["type"]	String	Format: dd-mm-yyyyThh:mm:ss	Date and time when the absolute usage accumulator for the selected Reporting Group expires for the subscriber or shared dataplan.  Possible value for "type" are as follows: <ul style="list-style-type: none"> <li>• volume for volume expiration</li> <li>• time for time expiration</li> </ul>



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. hasExpired["type"]</code>	bool	true false (1)	<p>Indicates if a prepaid Reporting Group expiration date and time has been reached.</p> <p>It returns <b>true</b> after expiration date (if expiration date is previous to endDate), and <b>false</b> otherwise.</p> <p>Possible values for “type” are as follows:</p> <ul style="list-style-type: none"><li>• volume for volume expiration</li><li>• time for time expiration</li></ul>
<code>AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. isActive["type"]</code>	bool	true false	<p>Indicates if the Reporting Group is active, that is:</p> <ul style="list-style-type: none"><li>• Between the subscription and the expiration date and time.</li><li>• AND between the startDate and endDate of the selected subscriber group</li></ul> <p>It returns:</p> <p><b>false:</b></p> <ul style="list-style-type: none"><li>• before startDate</li><li>• between startDate and subscription date</li><li>• after endDate</li></ul> <p><b>true</b> between startDate/subscription date and endDate.</p> <p>Possible values for “type” are as follows:</p> <ul style="list-style-type: none"><li>• volume for volume expiration</li><li>• time for time expiration</li></ul>



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. isLimitSurpassed["type"][n] (2)	bool	true  false	Indicates if the selected specified Reporting Group limit accumulated for the subscriber or shared dataplan has been surpassed.  "total" is used to indicate limits total traffic.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. remaining["type"][n] (2)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Remaining usage (limit - current usage accumulator if limit - current $\geq 0$ , 0 otherwise) of the Reporting Group selected for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Value of the Reporting Group usage accumulator selected for the subscriber and current IP session.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. isLimitSurpassed["type"][n] (2)	bool	true  false	Indicates if the specified Reporting Group limit accumulated for the subscriber and current IP session has been surpassed.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>





Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. remaining["type"][n]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Remaining usage (limit - current usage accumulator if limit - current $\geq 0$ , 0 otherwise) of the Reporting Group selected for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• u1Volume for volume uplink</li> <li>• d1Volume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
<code>AccessData. subscriber. receivedUsage. reportingGroup ["total"/ "reportingGroupName"]. usageType["type"]</code>	Integer	any	Usage reporting received for the subscriber for the indicated Reporting Group. <sup>(3)</sup>  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• u1Volume for volume uplink</li> <li>• d1Volume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>

(1) This tag always returns false for Postpaid subscriptions

(2) It may be used as array elements, in case intermediate limits are defined

(3) It only applies to traffic in PCC deployments.



Table 5 Fair Usage Related Tags, Complementary usage accumulators

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup ["total"/"reportingGroupName"]. counter["counterName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Value of the Reporting Group complementary usage accumulator selected for the subscriber or shared dataplan.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
AccessData. subscriber. accumulatedUsage. reportingGroup ["total"/"reportingGroupName"]. counter["counterName"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.
AccessData. subscriber. accumulatedUsage. reportingGroup ["total"/"reportingGroupName"]. counter["counterName"]. isLimitSurpassed["type"][n] (1)	bool	true  false	Indicates if the specified Reporting Group complementary limit for the subscriber has been surpassed.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>



Table 5 Fair Usage Related Tags, Complementary usage accumulators

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. counter["counterName"]. remaining["type"][n] (1)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Remaining usage (limit - current complementary usage accumulator) of the Reporting Group selected for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. counter["counterName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Value of the Reporting Group complementary usage accumulator selected for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. counter["counterName"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.



Table 5 Fair Usage Related Tags, Complementary usage accumulators

Tag	Return Type	Possible Values	Comments
<code>AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. counter["counterName"]. isLimitSurpassed["type"][n]</code> (1)	bool	true  false	Indicates if the specified Reporting Group limit accumulated for the subscriber and current IP session has been surpassed.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
<code>AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. counter["counterName"]. remaining["type"][n]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Remaining usage (limit - current complementary usage accumulator if limit - current $\geq 0$ , 0 otherwise) of the Reporting Group selected for the subscriber and current IP session.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
<code>AccessData. subscriber. accumulatedUsage. reportingGroup ["total"/ "reportingGroupName"]. counter["counterName"]. expiryDate["type"]</code>	string	Format:  dd-mm-yy yThh:mm:ss	Date and time when the complementary usage accumulator for the selected Reporting Group for the subscriber or shared dataplan is reset.  Possible values for “type” are as follows: <ul style="list-style-type: none"><li>• volume for volume expiration</li><li>• time for time expiration</li></ul>

(1) It may be used as array elements, in case intermediate limits are defined



In case Multiple service offering is configured, the tags can be specified indicating the particular group usage accumulators:

Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Value of the specified Reporting Group complementary usage accumulator (with the specified "groupName" <sup>(1)</sup> ) for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.
AccessData. subscriber. accumulatedUsage. reportingGroup ["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. expiryDate["type"]	string	Format:  dd-mm-yyy yThh:mm:ss	Date and time when the Reporting Group complementary usage accumulator (with the specified "groupName" <sup>(2)</sup> ) is reset for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• volume for volume expiration</li> <li>• time for time expiration</li> </ul>



Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
<code>AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. remaining["type"][n]</code> (7)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Remaining usage (limit - current complementary usage accumulator) of the specified Reporting Group complementary usage accumulator (with the specified "groupName" <sup>(3)</sup> ) for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
<code>AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Value of the Reporting Group usage accumulator (with the specified "groupName" <sup>(4)</sup> ) for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>
<code>AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. currentPercentage["type"]</code>	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.



Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. expiryDate["type"]	String	Format: dd-mm-yyy yThh:mm:ss	Date and time when the absolute accumulated usage for the Reporting Group (with the specified "groupName" <sup>(5)</sup> ) expires for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• volume for volume expiration</li> <li>• time for time expiration</li> </ul>
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. hasExpired["type"]	String	true false (6)	It returns <b>true</b> after expiration date (if expiration date is previous to endDate), and <b>false</b> otherwise.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• volume for volume expiration</li> <li>• time for time expiration</li> </ul>
AccessData. subscriber. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. remaining["type"][n] (7)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Remaining usage (limit - current usage accumulator) of the Reporting Group (with the specified "groupName" <sup>(8)</sup> ) for the subscriber or shared dataplan.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>



Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Value of the specified Reporting Group complementary usage accumulator, (with the specified "groupName" <sup>(9)</sup> ) for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. currentPercentage["type"]	Integer	0-100	Same as previous, but value is expressed as percentage of the corresponding limit.
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. counter["counterName"]. remaining["type"][n] (7)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Remaining usage (complementary limit - current usage accumulator) of the specified Reporting Group usage accumulator (with the specified "groupName" <sup>(10)</sup> ) for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>





Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. current["type"]	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes or seconds is lost.	Value of the Reporting Group usage accumulator (with the specified "groupName" <sup>(11)</sup> ) for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"> <li>• ulVolume for volume uplink</li> <li>• dlVolume for volume downlink</li> <li>• bidirVolume for volume bidirectional</li> <li>• time for time</li> </ul>



Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
<code>AccessData. subscriber. session. accumulatedUsage. reportingGroup ["total"/ "reportingGroupName"]. group["groupName"]. currentPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.
<code>AccessData. subscriber. session. accumulatedUsage. reportingGroup["total"/ "reportingGroupName"]. group["groupName"]. remaining["type"][n]</code> (7)	Integer	Kilobytes (volume) or minutes (time) as a whole number.  Precision of bytes and seconds is lost.	Remaining usage (limit - current usage accumulator) of the Reporting Group (with the specified "groupName" <sup>(12)</sup> ) for the subscriber and current IP session.  Possible values for "type" are as follows: <ul style="list-style-type: none"><li>• ulVolume for volume uplink</li><li>• dlVolume for volume downlink</li><li>• bidirVolume for volume bidirectional</li><li>• time for time</li></ul>

(1) dataplan

(2) dataplan

(3) dataplan

(4) dataplan

(5) dataplan

(6) This tag always returns false for Postpaid subscriptions

(7) It may be used as array elements, in case intermediate limits are defined

(8) dataplan

(9) dataplan

(10) dataplan

(11) dataplan

(12) dataplan

## 13.1 Fair Usage Policy Tags in Dynamic Group Selection Policies

This chapter contains considerations to take into account when using at the same time Dynamic Group Selection and fair usage policy tags (in any of the SAPC policy types).



Table 7 Fair Usage Policy Tags in Dynamic Group Selection Policies

Tag	Comments
<code>&lt;...&gt;.group["groupName"].&lt;...&gt;.remaining["type"]</code> <code>&lt;...&gt;.group["groupName"].&lt;...&gt;.current["type"]</code>	They should be used, including group name (dataplan) definition, when policies are required to detect when the limit is surpassed in cases of having the same Reporting Group associated to different groups.
<code>&lt;...&gt;.isLimitSurpassed["type"]</code>	It can be used when policies are required to detect when the limit is surpassed in cases of not having the same Reporting Group associated to different groups.
<code>&lt;...&gt;.group["groupName"].&lt;...&gt;.hasExpired</code>	It should be used, including group name (dataplan) definition, when policies are required to detect an expired Prepaid Reporting Group in cases of having the same Reporting Group associated to different groups.





# Glossary

**QoS**

Quality of Service

**SAPC**

Ericsson Service-Aware Policy Controller

**SASN**

Ericsson Service Aware Support Node





## Reference List

### Ericsson Documents

- [1] Configuration Guide for Access and Charging Control (Gx)
- [2] Managed Object Model (MOM)