

Configuration Guide for Fair Usage

Ericsson Service-Aware Policy Controller

USER GUIDE

Copyright

© Ericsson España, S.A. 2017, 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

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

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Configuration and Provisioning Overview	1
1.1	Other Conventions	2
2	Configuration Prerequisites	5
3	Configure Gx PCEFs for Usage Reporting	7
4	Configure Event-Triggers	9
5	Provision Fair Usage Profile	11
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
6	Provision Shared Subscriber Plans	27
7	Configure Quota Rollover	31
8	Configure Policy Control Based on Fair Usage	33
9	Provision Accumulation Policies	37
9.1	Multiple Usage Accumulators Based on Conditions	38
10	Usage Reporting Based on Multiple Gx	41
11	Monitor Usage Accumulators	45
11.1	Monitor Subscriber Usage Accumulators	45
11.2	Monitor Shared Dataplan Usage Accumulators	45
12	Configure Aggregable Dataplan	47
13	Provision Durations for Stackable Dataplan	49
14	Configuration Examples for Use Cases	51
14.1	Basic Turbo Button Based on Temporary Subscriber Groups	51
14.2	Advanced Turbo Button with Dynamic Group Selection Based on Fair Usage	54



14.3	Shared Subscriber Plans Use Cases	57
14.3.1	Shared and Individual Limits for Members	57
14.3.2	Category Limits for Shared Dataplan Members	61
14.3.3	Monitoring Shared Subscriber Plan Members	65
14.3.4	Combining Individual Plans and Shared Subscriber Plan	68
14.3.4.1	Individual and Shared Plans for Different Reporting Groups	68
14.3.4.2	Plans for the Same Reporting Group Prioritizing the Individual Plan	70
14.3.4.3	Plans for the Same Reporting Group Prioritizing the Shared Subscriber Plan	72
14.4	Aggregable Dataplan for Fair Usage Policies	73
14.5	Stackable Dataplan	75
15	Appendix A. Fair Usage Policy Types	79
16	Appendix B. Fair Usage Policy Tags	81
16.1	Fair Usage Policy Tags in Dynamic Group Selection Policies	95



1 Configuration and Provisioning Overview

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

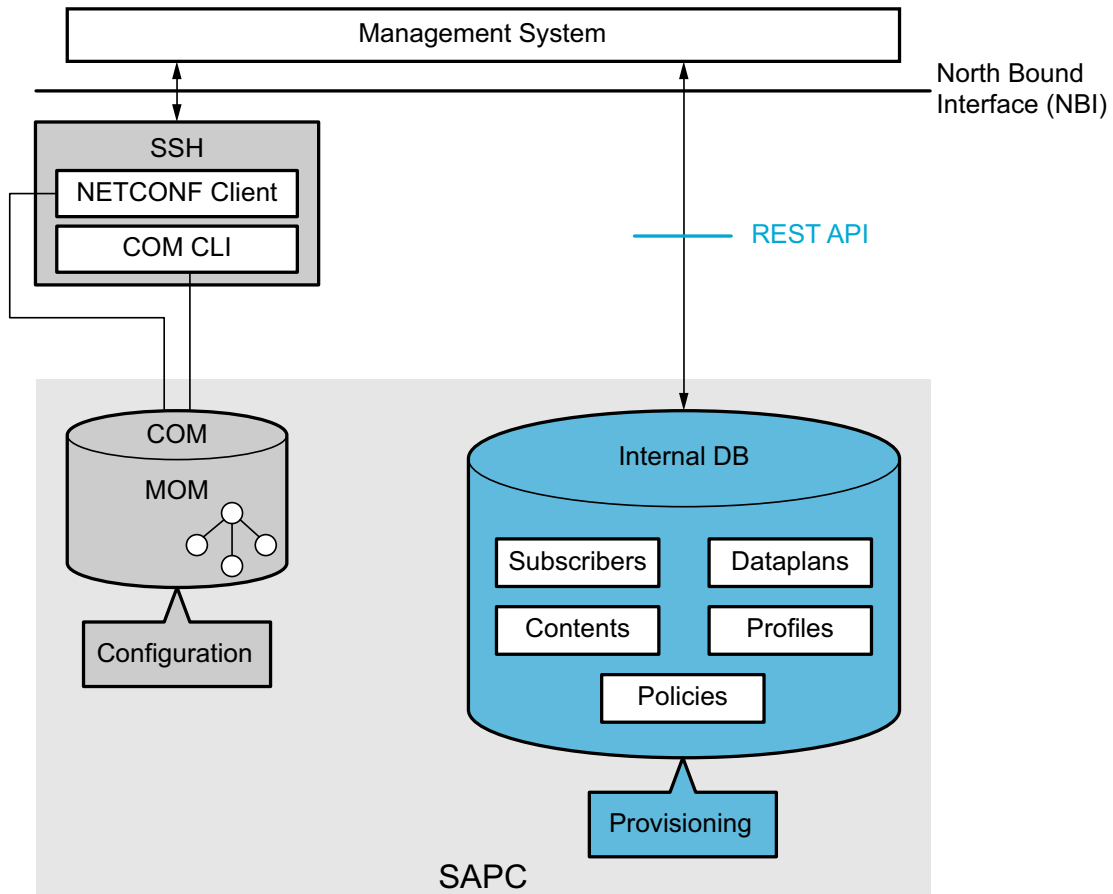


Figure 1 Configuration and Provisioning Overview

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 extensive guide for configuring all possibilities related to Fair Usage in the SAPC.

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

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



1.1 Other Conventions

This document refers to some configuration and provisioning data.

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

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

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

Example: set `enableReauthsOnSubsChange` attribute in class `AppConfig`.

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

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

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

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

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

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

HTTP-Operation /resource-URI
{json content} where /resource-URI is the relative URI from the SAPC provisioning base URI detailed in [Provisioning REST API](#).

Example:



```
PUT /dataplan/Gold
{ "dataplanName" : "Gold",
  "subscribedContents" : [{"contentName" : "HTTP_Streaming",
                           "redirect" : false}]
}
```

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





2 Configuration Prerequisites

Before configuring the SAPC in an operational network, make sure that:

- CBA Components are installed
- The SAPC product software is installed
- The SAPC user performing configuration changes has thorough knowledge 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, the subscriber data in the SAPC, to layered solution, the 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 1 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within <code>usageLimits</code>	Comments
Reporting Group	<code>"name": "reportingGroupName"</code>	Identifies the Reporting Group 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.
Usage Limits Aggregable	<code>"aggregable" : false</code>	Indicates whether the usage limits of this Reporting Group are aggregable. If <code>true</code> , the usage limits of this Reporting Group can be aggregated with other aggregable Reporting Groups, with the same name, of the subscriber groups associated with the same subscriber. The default value is <code>false</code> . If empty, the SAPC uses the default value.



Table 1 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within usageLimits		Comments
Reporting Granularity	"reportingLevel" : "totalTraffic/perReportingGroup"		Usage limits can apply at total traffic level or Reporting Group level, that is, a service or group of services.
	For total traffic	"name": "reportingGroupTotalTraffic" "reportingLevel": "totalTraffic"	The reportingGroupName corresponds with the usage reported under Monitoring-Key AVP in agreement with the PCEF within Usage-Monitoring-Information AVP containing Usage-Monitoring-Level AVP set to SESSION_LEVEL.
	Reporting Group	"reportingLevel": "perReportingGroup" "name": "reportingGroup"	The reportingGroupName corresponds with the usage reported under Monitoring-Key AVP in agreement with the PCEF within Usage-Monitoring-Information AVP containing Usage-Monitoring-Level AVP set to PCCRULE_LEVEL.
Subscription Type	Postpaid See Section 5.1 on page 14	"subscriptionType": "postpaid"	Changes in Postpaid or Prepaid Subscription of a Subscriber: 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 absolute or complementary limits 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 1 Configurable Elements in Fair Usage Profile

Fair Usage Element	Configurable attribute within usageLimits		Comments
Life Cycle of usage accumulators ⁽²⁾	Subscription Date	"subscriptionDate": "dd-mm-yyyyThh:mm:ss"	<p>When no 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>Flexible Billing Cycles: 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 <code>startDate</code>, then subscription date takes its value from the <code>startDate</code> field.</p>
	Reset Period	<pre>"resetPeriod":{ "volume":"monthly"/ "<nr> days"/ "<nr> hours"/ "monthly day <1-31> hh:mm"/ "weekly day <Monday-Sunday> hh:mm"/ "daily hh:mm", "time": "monthly"/ "<nr> days"/"<nr> hours"/ "monthly day <1-31> hh:mm"/ "weekly day <Monday-Sunday> hh:mm"/ "daily hh:mm" },</pre>	



Table 1 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 different policy control actions are required: set limits to value 0.

(2) Considerations with Daylight Saving Time.

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.

In case of subscriber groups with different priority, there is no conflict, as the SAPC selects the Fair Usage Profile defined in the subscriber group with higher priority.

5.1 Provision Postpaid Subscriptions

Here is 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 is 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" : ""
    }
  ],
  "pccRuleName" : "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.

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, expiry date or both. Account refill is also possible.

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 with unknown subscriber.

Next is 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 when 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 the `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].isActive["<type>"]` and `AccessData.subscriber.accumulatedUsage.reportingGroup["total"].hasExpired["<type>"]` tags properly 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 or Recharge

To refill or recharge the subscription, modify the subscriber subscriptionDate, the current accumulated consumed usage is then reset, and if needed, update the usageLimits attribute, 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 the usageLimits attribute.

Table 2 Refill or 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 and add the difference between the previous limit and the current accumulator. See Section 11 on page 45 for more details on usage 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 is 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. For example, if the subscriber had 0.5 Gbytes unspent as remaining quota, and the operator wants to add them to the previous limit, which is 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 takes the characteristics of the normal group automatically, that is without OAM intervention.



```

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, and endDate the dates corresponding to the period where the subscriber has the possibility of buying the prepaid voucher from 01-07-2020 until 31-07-2020.

Refill or Recharge

In this case, the refill for the voucher can also be done by modifying the attribute startDate within the 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 9 on page 37.

Figure 2 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 how to set different usage limits at different reset periods for a Reporting Group. It is possible to set a volume limit of 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.

Figure 3 shows a graphical summary of the configuration:

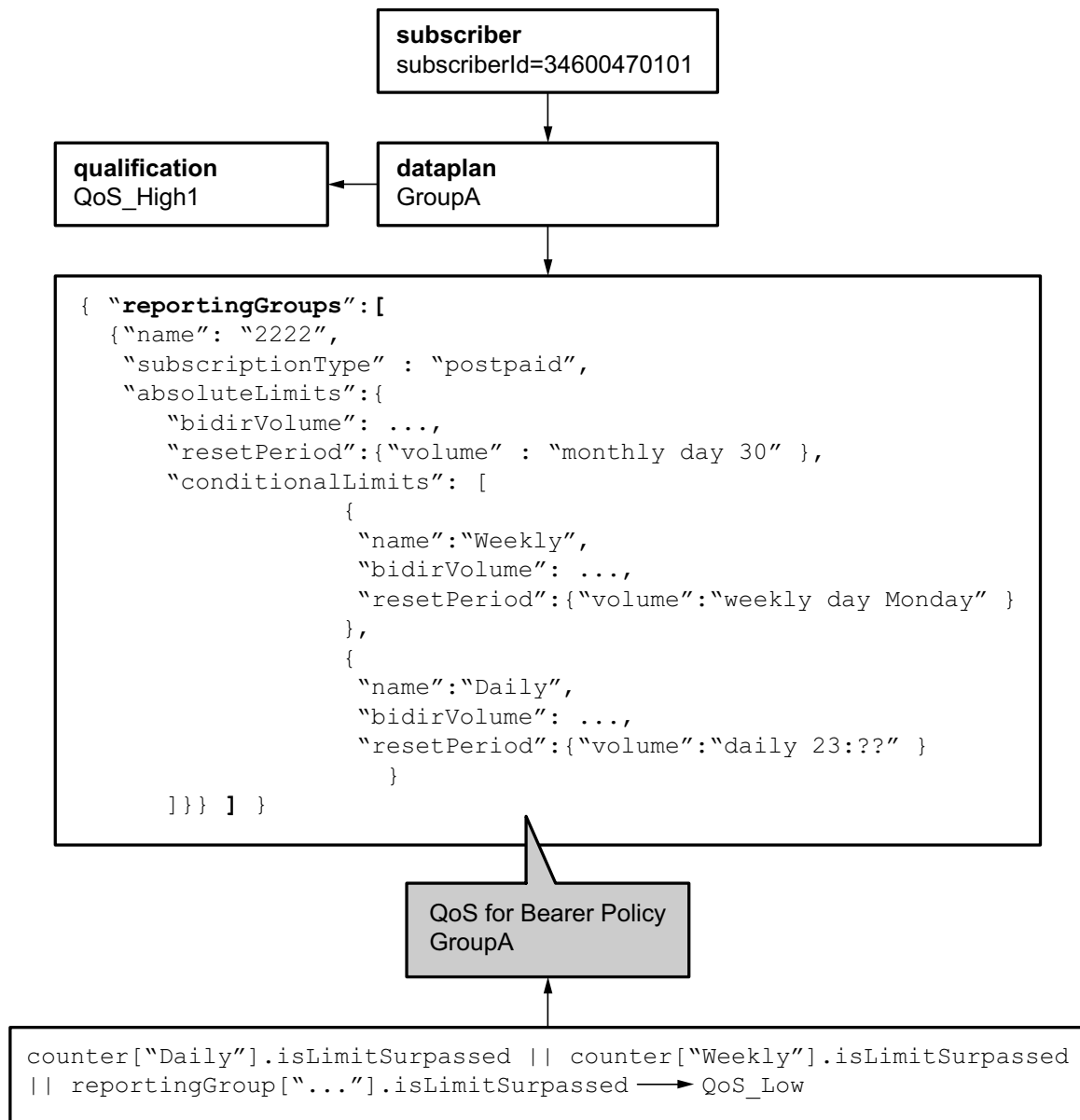


Figure 3 Complementary Limits with different reset periods

```
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 usage of the subscriber 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 owing 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, allowing several subscribers to report their 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.

Figure 4 shows the relations of the shared dataplan related data:

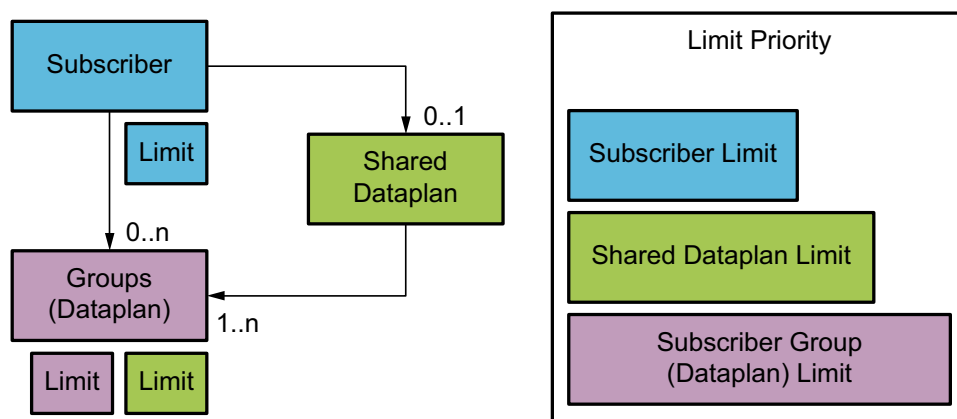


Figure 4 Shared Dataplan Relations

The relations shown 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.

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 the 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 14.3 on page 57.

The usage limits that are shared among the members of the shared dataplan are the **absolute limits**, including both general and multiple counters 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, or the 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:
Session accumulation.

Next example shows a shared subscriber plan:

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

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

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

PUT /subscriber/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 Quota Rollover

Set the `useRolloverFirst` attribute in the `usageLimit` object, and the `rolloverLimit` attribute in the `absoluteLimit` or `conditionalLimit` object to activate the transfer of remaining data to the next period for a subscriber or shared data plan. The attributes apply to volume and time usage types for the selected limit types of postpaid subscriptions.

They do not apply to complementary or session limits.

<code>useRolloverFirst</code>	<ul style="list-style-type: none"> • This attribute can only be applied at Reporting Group level • If set to <code>true</code>, it indicates that the rollover quota is consumed before the quota corresponding to the regular usage limit • Default value is set to <code>true</code>⁽¹⁾
<code>rolloverLimit</code>	<ul style="list-style-type: none"> • Applicable for absolute and conditional limits • Maximum percentage of data usage limit, which can be transferred to the next billing period if unconsumed • Values: 0-100. Default value is 0 which means that quota rollover is not transferred to the next period⁽¹⁾

(1) Changing the attributes during a billing period has no effect until the beginning of the next period.

The example describes a dataplan for postpaid subscriptions with an absolute limit of 100%. The configuration includes an intermediate limit set at 80% as a usage limit for the bidirectional traffic volume of `GoldWithRollover` subscribers. The usage limit for the period is 10 GB with a monthly reset, which is managed in 2 MB slices. The rollover configuration indicates that the rollover quota is 100% of the remaining quota and it is consumed before the dataplan in the next period. The example also describes a policy defined to send an end user notification when the amount of rollover quota defined for the billing period is depleted.



```
PUT /dataplan/GoldWithRollover
```

```
{
  "dataplanName" : "GoldWithRollover",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : [ "80%", 10485760],
        "rolloverLimit" : "100%",
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "subscriptionType" : "postpaid",
      "useRolloverFirst" : true,
      "sliceVolume" : 20480
    }
  ]
}
```

```
PUT /rules/r_Notif_RolloverSurpassed
```

```
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"total\"]').isRolloverSurpassed",
  "outputAttributes" :
  [
    {
      "attrName" : "notification",
      "attrValue" : "\"You have consumed all the data rolled over from previous period.\"\"",
      "result" : "permit"
    }
  ],
  "ruleName" : "r_Notif_RolloverSurpassed"
}
```

```
PUT /policies/p_Notif_RolloverSurpassed
```

```
{
  "policyName" : "p_Notif_RolloverSurpassed",
  "ruleCombiningAlgorithm" : "all-permit",
  "rules" :
  [
    "r_Notif_RolloverSurpassed"
  ]
}
```

```
PUT /dataplan/GoldWithRollover/locators/resources/any/contexts/notification
```

```
{
  "policies" : [ "p_Notif_RolloverSurpassed" ]
}
```

Example 11 Configuration for Quota Rollover



8 Configure Policy Control Based on Fair Usage

When the subscriber reaches the provisioned limits, the SAPC can deny access to the service, change the bandwidth setting, or even apply different ratings depending on the provisioned policies.

This section shows how to configure the SAPC to act when usage limits are surpassed.

Next is a typical example to authorize a service based on Usage Reporting limits:

```
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 12 Configuration for Service Authorization based on Usage Control

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

It is also possible to authorize 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 is an example of how to configure 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 /dataplanes/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 13 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 is 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 /locators/resources/Internet/contexts/static-access
{
  "policies" : [ "pQualifyPrepaid" ]
}

```

Example 14 Policies for Prepaid





9 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>.

— For **Global policy locator**:

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

— For **Subscriber group locator**:

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

— For **Subscriber locator**:

```
/subscribers/ <subscriberId>/locators/resources/reporting-group
/contexts/accumulation or /subscribers/ <subscriberId>/locators/r
esources/<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 is an example for a global policy which is 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 15 Policies for Accumulation



9.1 Multiple 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/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 is 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 16 Conditional Accumulation





10 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 17 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 used to accumulate the consumed usages in the SAPC.

When the bidirectional total traffic consumed usage reaches the limit, as shown 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`.

The most restrictive condition for the rules within the same policy has to be configured first, as permit overrides combination algorithm is implicitly used. In this case, it is the condition about the surpassed limit.

Configuration regarding other controls, such as IP session Access Control, Service Access Control, or 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 \(Gx\)](#).





11 Monitor Usage Accumulators

11.1 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.

11.2 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.





12 Configure Aggregable Dataplan

To aggregate usage limits, configure aggregable Reporting Groups in the subscriber groups associated with the subscriber.

Set the optional aggregable attribute to `true` for the Reporting Groups with the same name, also `total` or `no name`.

Note: The SAPC returns a failure for the following configurations. See [Provisioning REST API](#) for details.

- `"aggregable" : true` and `"subscriptionType" : "prepaid"` are provisioned.
- `"aggregable" : true` and intermediate volume limits not expressed as percentage.

Intermediate time limits are not applicable for usage limits aggregation.

In Example 18, the Reporting Group **100** can be aggregated with other aggregable Reporting Groups named **100** of the subscriber groups associated with the same subscriber.

```
PUT /dataplan/AggregateGold
{
  "dataplanName" : "AggregateGold",
  "usageLimits" :
  [
    {
      "name" : "100",
      "aggregable" : true,
      "absoluteLimits" :
      {
        "bidirVolume" : "50%", 1048576,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      },
      "description" : "Total traffic",
      "subscriptionType" : "postpaid"
    }
  ]
}
```

Example 18 Configuration of Aggregable Reporting Groups

Ericsson recommends configuring the following data in the subscriber group with the highest priority, among all the subscriber groups containing the aggregable Reporting Groups with the same name:

- Intermediate volume limits of absolute limits.
- All non-aggregable attributes, such as attributes of usage limits except `absolute ulVolume`, `dlVolume`, `bidirVolume` limits, and time limit of fair usage profiles.



Dynamic Group Selection policies are not applicable for the subscriber groups containing aggregable Reporting Groups.

Usage limits aggregation does not apply to Reporting Groups provisioned at subscriber level or in shared subscriber plans.



13 Provision Durations for Stackable Dataplans

To assign a subscriber group to a subscriber in multiple periods of time, use the `durations` attribute in the `subscribers` URI in the provisioning REST API.

Note: Multiple instances apply to the subscriber groups containing only absolute usage limits with or without intermediate limits in the Fair Usage profile, so do not provision complementary, session, or conditional limits in any Reporting Group of such subscriber groups.

In Example 19, the `PrepaidMultipleInstances` subscriber group is assigned to the subscriber between `06-06-2020` and `06-07-2020`, between `06-07-2020` and `06-08-2020`, and between `24-08-2020` and `24-09-2020`.

```
PUT /subscribers/38500322020
{
  "dataplans" :
  [
    {
      "dataplanName" : "PrepaidMultipleInstances",
      "priority" : 1,
      "durations" :
      [
        {"duration" : "06-06-2020, 06-07-2020"},
        {"duration" : "06-07-2020, 06-08-2020"},
        {"duration" : "24-08-2020, 24-09-2020"}
      ]
    }
  ],
  "subscriberId" : "38500322020",
  "trafficIds" : [ "898345000" ]
}
```

Example 19 Provisioning of Durations for a Prepaid Dataplan

Note:

- The `durations` attribute can be configured only for prepaid subscriptions.
- The `durations` attribute cannot be configured together with `startDate` or `stopDate`, or both.
- In durations, start dates must be earlier than the corresponding stop dates.
- The same durations are not allowed for the same subscriber group association.





14 Configuration Examples for Use Cases

14.1 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 a higher bandwidth with different limits, for example 250 MB at 3 Mbps, from 20:00 to 22:00 of the 1 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.

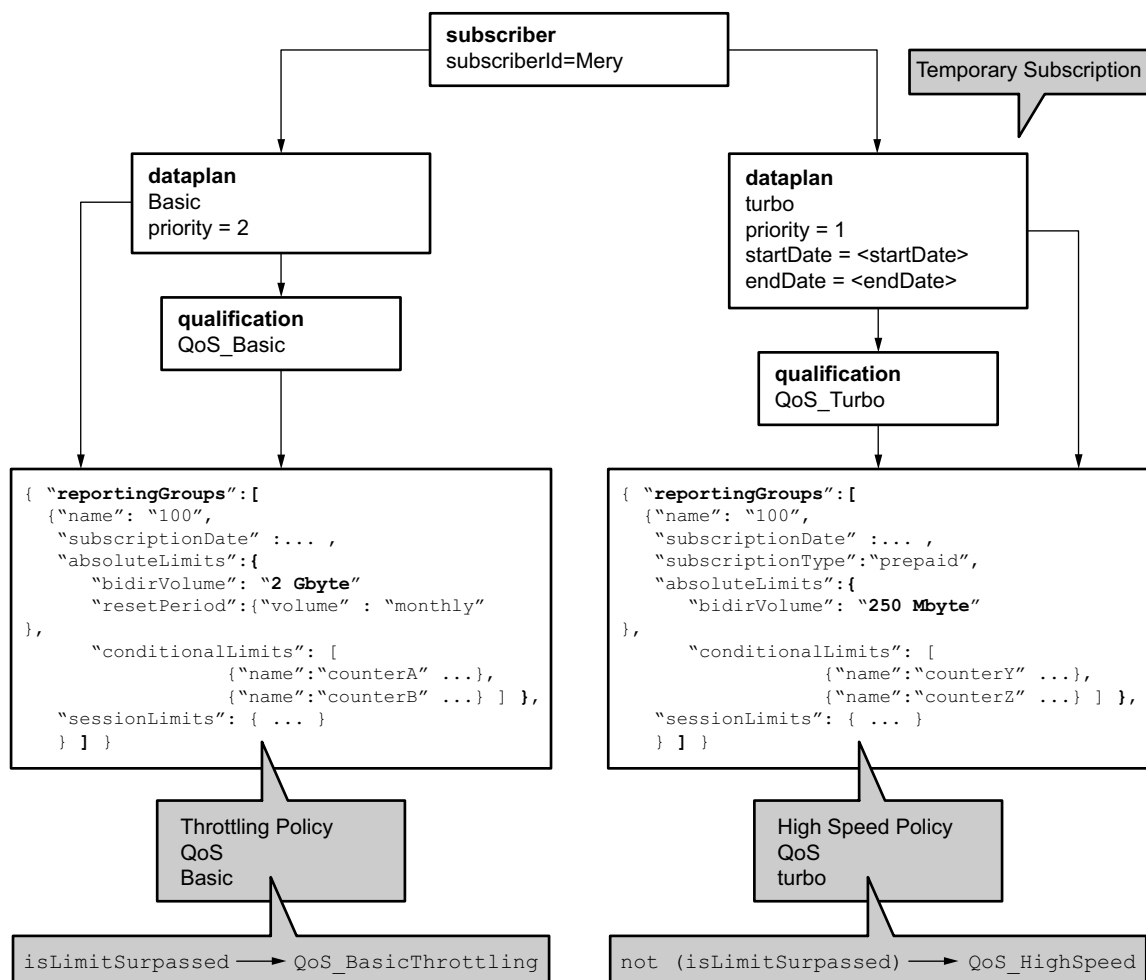


Figure 5 Basic Turbo Button based on Temporary Subscriber Groups



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

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

PUT /dataplan/Basic
{
  "dataplanName" : "Basic",
  "staticQualification" :
  {
    "maxBearerQosProfileId" : "QoS_Basic"
  },
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
```




```

        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
            "volume" : "monthly"
        }
    },
    "description" : "Peer to Peer",
    "name" : "100"
}
]
}

PUT /dataplan/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 20 Basic Turbo Button based on Temporary Subscriber Groups

In the example, 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 1 January.

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

- 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 sped up, so the Bearer QoS profile applied is QoS_Turbo.

14.2 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 first. 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.

Figure 6 shows a graphical sketch of the needed configuration:

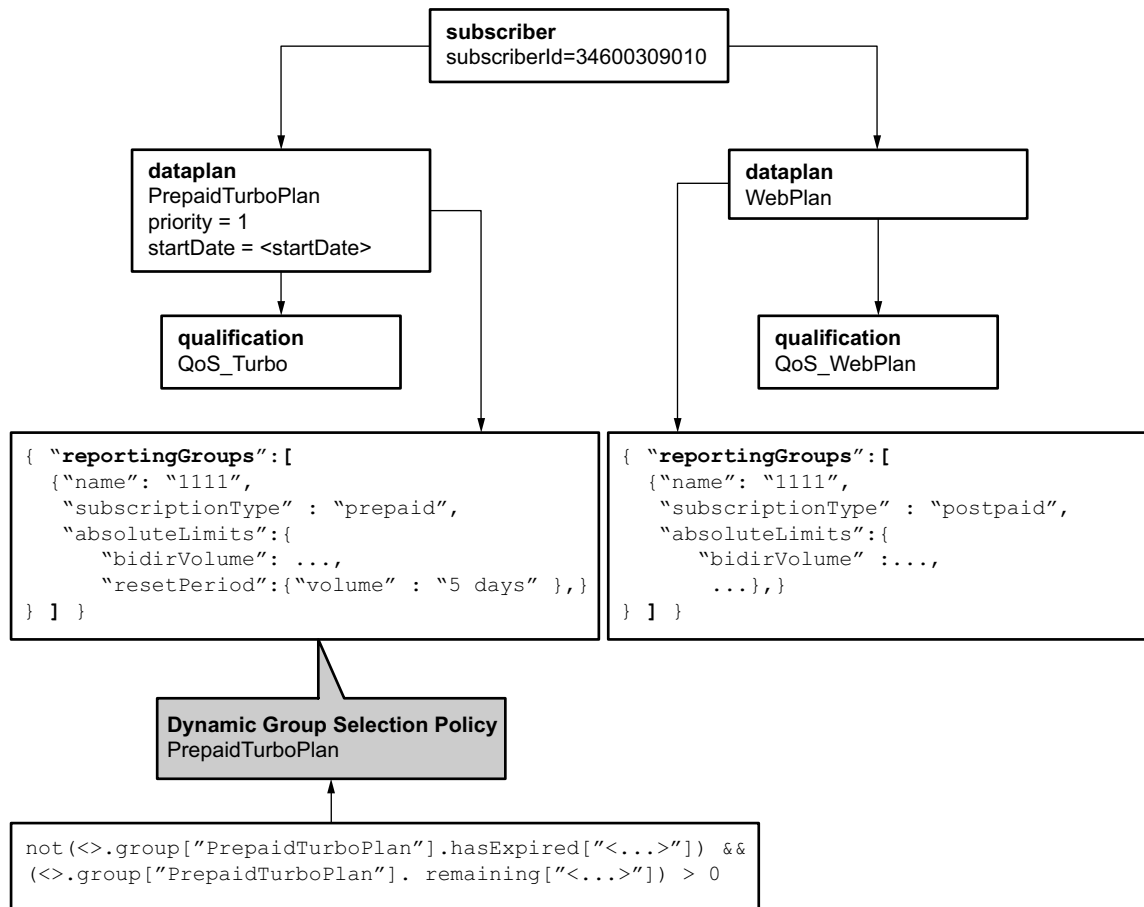


Figure 6 Advanced Turbo Button with Dynamic Group Selection Based on Fair Usage

```

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 21 Advanced Turbo Button with Dynamic Group Selection based on Fair Usage



In the example, 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 06-01-2020. 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 16.1 on page 95.

14.3 Shared Subscriber Plans Use Cases

14.3.1 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.

Figure 7 shows a figure summarizing of the configuration:

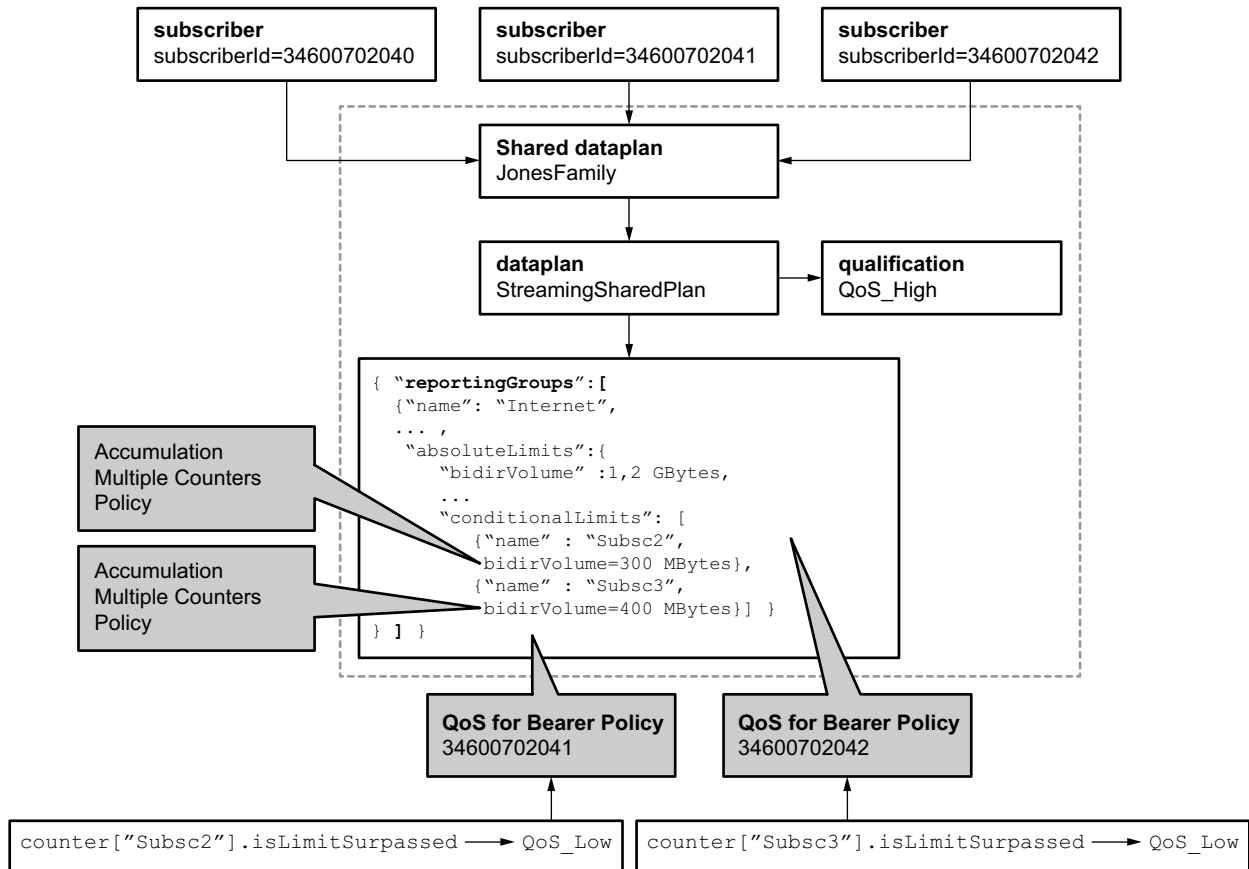


Figure 7 Shared and Individual Limits for Members

```
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 22 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 owing to accumulation policies, and the QoS profile is downgraded to QoS_Low.

14.3.2 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 category of the member. To do so, use the `operatorSpecificInfos` attribute of the subscriber to store the type of the member. 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 benefit from the Shared Subscriber Plan.

Figure 8 shows a graphical sketch of the configuration:

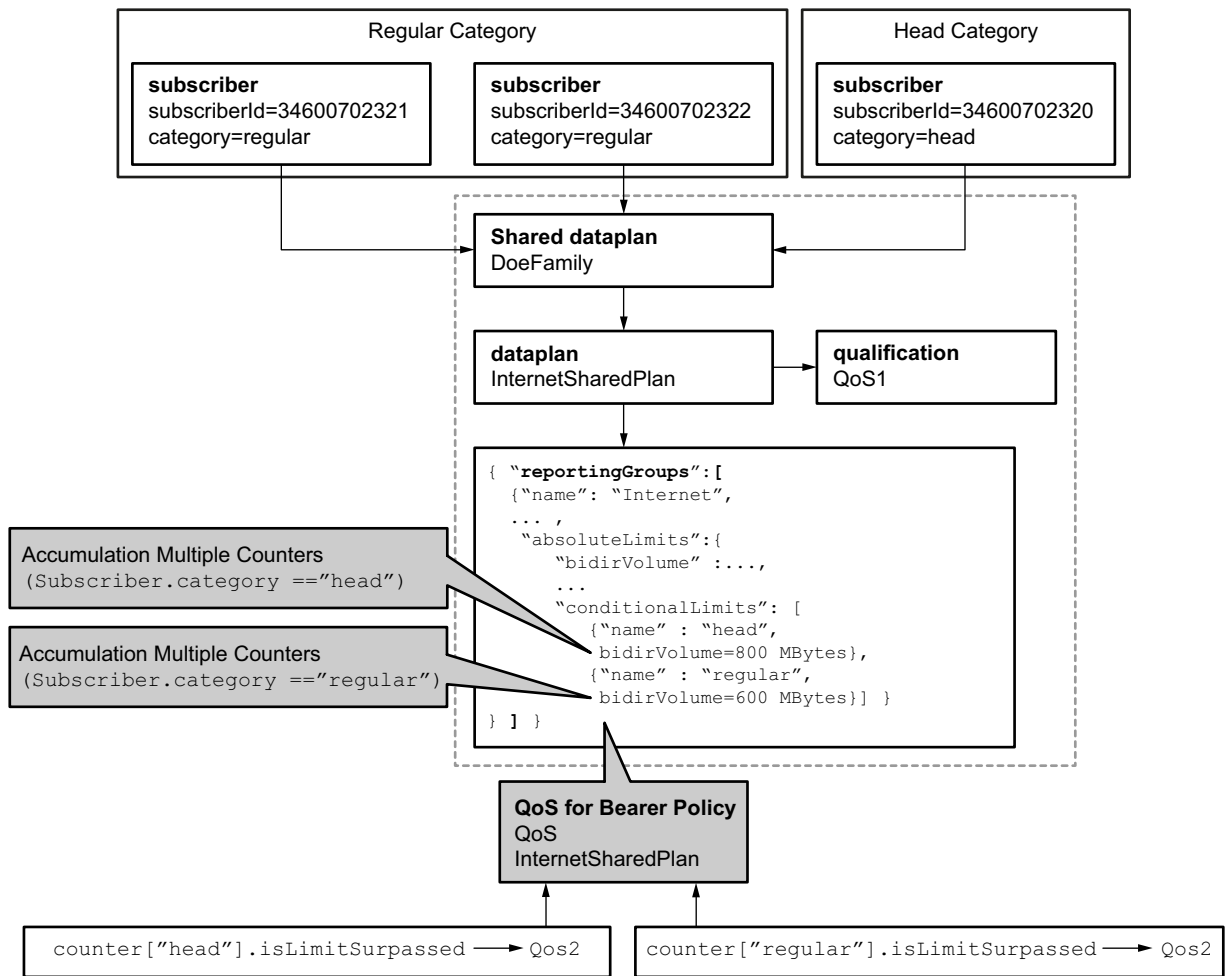


Figure 8 Category Limits for Shared Dataplan Members

```

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

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

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

PUT /dataplan/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 23 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 and 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.

14.3.3 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-dataplan/SmithFamily
{
  "dataplan" :
```



```
[
  {
    "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 24 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 25 Configuration to monitor categories of members

14.3.4 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.

14.3.4.1 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.

Figure 9 shows a graphical sketch of the configuration:

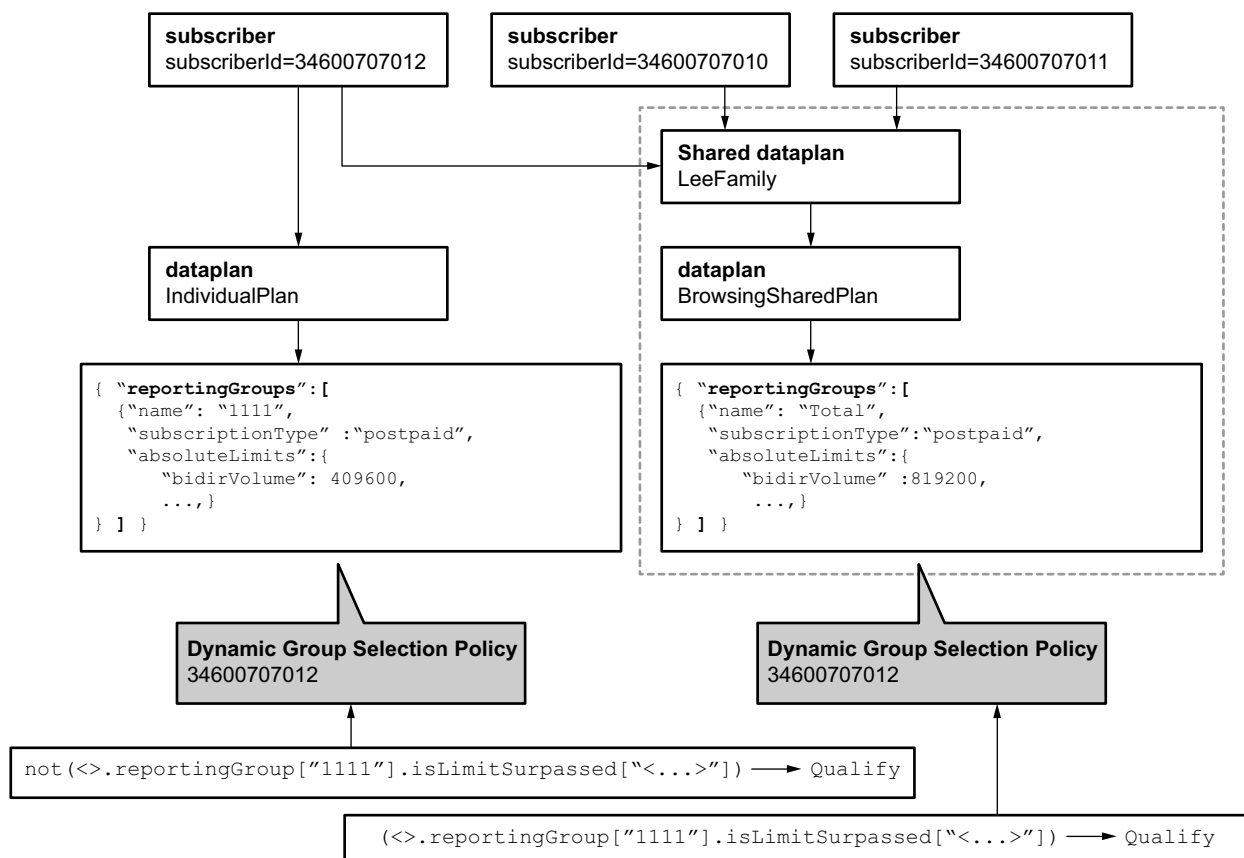


Figure 9 Individual and Shared Plans for Different Reporting Groups

```

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-dataplans/LeeFamily
{
  "dataplans" :
  [
    {
      "dataplanName" : "BrowsingSharedPlan"
    }
  ],
  "sharedDataplanId" : "LeeFamily"
}

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

PUT /subscribers/34600707011
{
  "sharedDataplan" : "LeeFamily",
  "subscriberId" : "34600707011"
}

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

Example 26 Combination of Individual and Shared Subscriber Plans

In the example, the members of LeeFamily, subscribers 1, 2 and 3, have Total Reporting Group defined. Subscriber 34600707012 also has 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 16.1 on page 95.

14.3.4.2

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 27 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 16.1 on page 95.



14.3.4.3 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 or 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 28 Combination of Individual and Shared Data plan, prioritizing Shared Data Plan



The configuration 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 16.1 on page 95.

14.4 Aggregable Dataplans for Fair Usage Policies

In this scenario, the subscriber John belongs to two subscriber groups. The Reporting Groups named **100** of the two subscriber groups can be aggregated. The SAPC performs fair usage control for the aggregated usage limits.

Figure 10 is a graphical sketch of the configuration.

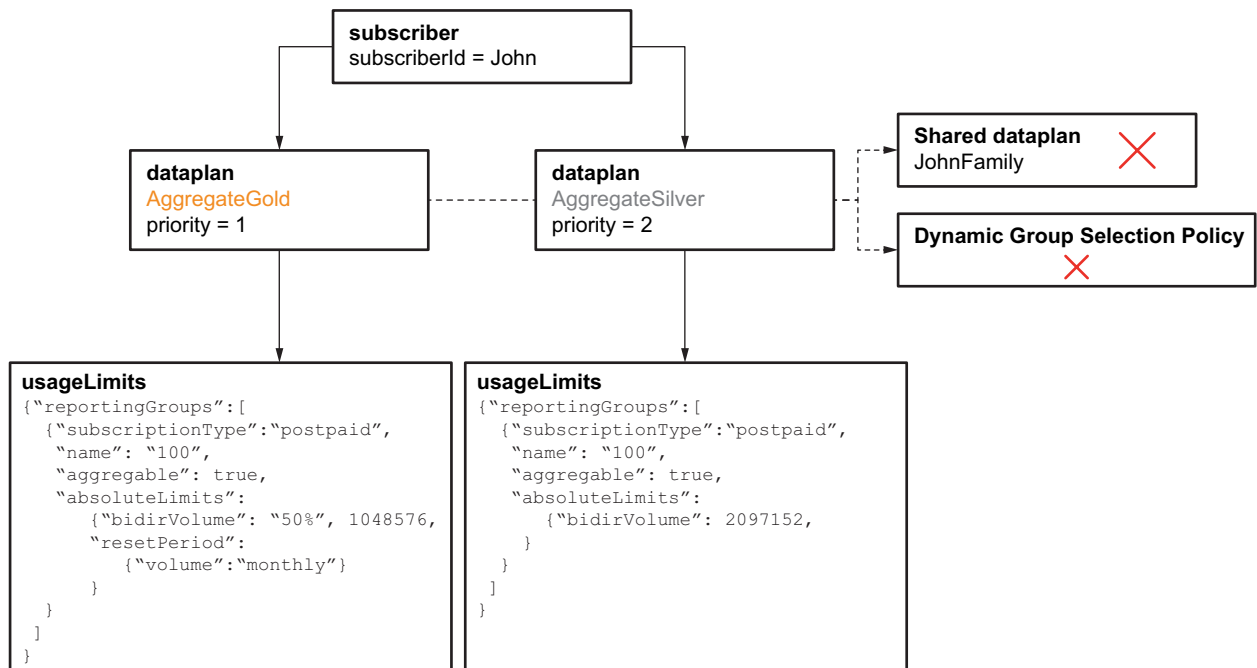


Figure 10 Configuration of Usage Limits Aggregation

Example 29 provisions two subscriber groups AggregateGold and AggregateSilver for which the usage limits can be aggregated.



```

PUT /dataplan/AggregateGold
{
  "dataplanName" : "AggregateGold",
  "usageLimits" :
  [
    {
      "name" : "100",
      "aggregable" : true,
      "absoluteLimits" :
      {
        "bidirVolume" : ["50%", 1048576],
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      }
    },
    {
      "description" : "Total traffic",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /dataplan/AggregateSilver
{
  "dataplanName" : "AggregateSilver",
  "usageLimits" :
  [
    {
      "name" : "100",
      "aggregable" : true,
      "absoluteLimits" :
      {
        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
          "volume" : "monthly"
        }
      }
    },
    {
      "description" : "Total traffic",
      "subscriptionType" : "postpaid"
    }
  ]
}

PUT /subscriber/John
{
  "dataplan" :
  [
    {
      "dataplanName" : "AggregateGold",
      "priority" : 1
    },
    {
      "dataplanName" : "AggregateSilver",
      "priority" : 2
    }
  ],
  "smsDestinations" : [ "+3461230170403" ],
  "subscriberId" : "John"
}

```

Example 29 Configuration of Usage Limits Aggregation

Example 30 shows a notification policy configuration based on the usage accumulation of the aggregated limits.



```

PUT /rules/rAggregatedLimitSurpassedPercentage
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.reportingGroup[\"100\"]').group[\"_Agg
  "outputAttributes" :
  [
    {
      "attrName" : "notification",
      "attrValue" : "\"You have consumed half of the usage limits.\\"",
      "result" : "permit"
    }
  ],
  "ruleName" : "rAggregatedLimitSurpassedPercentage"
}

PUT /policies/pAggregatedNotifications
{
  "policyName" : "pAggregatedNotifications",
  "ruleCombiningAlgorithm" : "all-permit",
  "rules" :
  [
    "rAggregatedLimitSurpassedPercentage"
  ]
}

PUT /dataplan/AggregateGold/locators/resources/any/contexts/notification
{
  "policies" : [ "pAggregatedNotifications" ]
}

```

Example 30 Configuration of a Notification Policy for Aggregated Usage Limits

14.5 Stackable Dataplan

In this scenario, the `PrepaidGoldMultiInstances` subscriber group is assigned to the subscriber in multiple periods of time. The subscriber can use multiple times 2 GB traffic from the subscriber group. The related notifications can be sent to the subscriber.

Example 31 provisions the `PrepaidGoldMultiInstances` subscriber group and assigns it to the subscriber.



```

PUT /dataplan/MultiInstancesGold
{
  "dataplanName" : "MultiInstancesGold",
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2097152,
        "resetPeriod" :
        {
          "volume" : "7 days"
        }
      },
      "name" : "100",
      "description" : "Total traffic",
      "subscriptionType" : "prepaid"
    }
  ]
}

PUT /subscribers/34600003857
{
  "dataplan" :
  [
    {
      "dataplanName" : "MultiInstancesGold",
      "priority" : 1,
      "durations" :
      [
        { "duration" : "06-06-2020, 06-07-2020"},
        { "duration" : "06-07-2020, 06-08-2020"},
        { "duration" : "24-08-2020, 24-09-2020"}
      ]
    }
  ],
  "smsDestinations" : [ "+34600003857" ],
  "subscriberId" : "34600003857",
  "trafficIds" : [ "778857000" ],
  "usageLimits" :
  [
    {
      "absoluteLimits" :
      {
        "bidirVolume" : 2621440,
        "resetPeriod" :
        {
          "volume" : "30 days"
        }
      },
      "description" : "Total traffic",
      "subscriptionDate" : "01-11-2020",
      "subscriptionType" : "prepaid"
    }
  ]
}

```

Example 31 Association of Subscriber and Subscriber Group

Example 32 configures a notification sent when the subscriber starts to use a new instance and it informs the start date and end date of the instance.



```

PUT /rules/rNewInstanceStart
{
  "condition" : "(AccessData.subscriber.accumulatedUsage.group[\"MultiInstancesGold\"]').newIns
  "outputAttributes" :
  [
    {
      "attrName" : "notification",
      "attrValue" : "strcat(\"You are using a new voucher\", AccessData.subscriber.accumu
      "result" : "permit"
    }
  ],
  "ruleName" : "rNewInstanceStart"
}

PUT /policies/pNewInstanceStartNotifications
{
  "policyName" : "pNewInstanceStartNotifications",
  "ruleCombiningAlgorithm" : "all-permit",
  "rules" : [ "rNewInstanceStart" ]
}

```

Example 32 Configuration of Notification on New Instance Start

Example 33 is used to tell the subscriber that no other instance is available to be consumed when the last one of multiple instances associated with the subscriber group is used up.

```

PUT /rules/rInstanceUnavailable
{
  "condition" : "not(AccessData.subscriber.accumulatedUsage.group[\"MultiInstancesGold\"]').isI
  "outputAttributes" :
  [
    {
      "attrName" : "notification",
      "attrValue" : "\"You have no available voucher.\",",
      "result" : "permit"
    }
  ],
  "ruleName" : "rInstanceUnavailable"
}

PUT /policies/pInstanceUnavailableNotifications
{
  "policyName" : "pInstanceUnavailableNotifications",
  "ruleCombiningAlgorithm" : "all-permit",
  "rules" :
  [
    "rInstanceUnavailable"
  ]
}

PUT /dataplan/MultiInstancesGold/locators/resources/any/contexts/notification
{
  "policies" : [ "pInstanceUnavailableNotifications", "pNewInstanceStartNotifications" ]
}

```

Example 33 Configuration of Notification on Unavailable Instance





15 Appendix A. Fair Usage Policy Types

Table 3 shows the different policy types related to Fair Usage.

Table 3 Accumulation Related Policies

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
Conditional Accumulation for Multiple Counters	accumulation	reporting-group,<contentId>.<counterId> <reporting-group-id>.<counter-Id>	<subscriberId> <dataplanId>	-	Type I = Only policies, no qualification Used to accumulate on different counters





16 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.

When the policy tags are used for Stackable Dataplan, the values of the tags are based on the instance in use. The policy tags containing counter or session are not applicable for Stackable Dataplan.

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.

For all the tables, `rolloverFromPreviousPeriod`, `currentRollover`, `isRolloverSurpassed`, and `currentRolloverPercentage` tags are only applicable for postpaid Reporting Groups, but not applicable to complementary or session limits.

Attention!

If an intermediate limit is configured as a percentage of the final limit, the SAPC calculates the intermediate limit as $\text{percentage value} * (\text{usage limit} + \text{rollover quota})$.

Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	Value of the usage accumulator of the selected Reporting Group for the subscriber or shared dataplan. When Quota Rollover is enabled, the calculation of the value is: <code>usage accumulator + rollover usage accumulator</code> Possible values for type are as follows: <ul style="list-style-type: none"> • <code>ulVolume</code> for volume uplink • <code>dlVolume</code> for volume downlink • <code>bidirVolume</code> for volume bidirectional • <code>time</code> for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].currentPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].expiryDate["type"]⁽¹⁾</code>	String	Format: dd-mm-yyyyThh:mm:ss	Date and time when the absolute usage accumulator of the selected Reporting Group for the subscriber or shared dataplan expires. Possible values for type are as follows: <ul style="list-style-type: none">• volume for volume expiration• time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].hasExpired["type"]⁽¹⁾</code>	bool	true false ⁽²⁾	It indicates if a prepaid Reporting Group expiration date and time has been reached. It returns true after expiration date, if expiration date is before the endDate, and false otherwise. Possible values for type are as follows: <ul style="list-style-type: none">• volume for volume expiration• time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].isActive["type"]⁽¹⁾</code>	bool	true false	It indicates if the Reporting Group is active, that is: <ul style="list-style-type: none">• Between the subscription and the expiration date and time• AND between the startDate and endDate of the selected subscriber group It returns: false : <ul style="list-style-type: none">• before startDate• between startDate or subscription date• after endDate true between startDate or subscription date, and endDate Possible values for type are as follows: <ul style="list-style-type: none">• volume for volume expiration• time for time expiration



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].isLimitSurpassed["type"][n]⁽¹⁾⁽³⁾</code>	bool	true false	<p>It indicates if the usage limit of the selected Reporting Group for the subscriber or shared dataplan has been surpassed.</p> <p>The limit is surpassed if</p> $(\text{usage limit} - \text{usage accumulator}) \leq 0.$ <p>Calculation of limits when Quota Rollover is enabled:</p> <ul style="list-style-type: none"> For absolute limits, the limit is surpassed if $(\text{usage limit} + \text{rollover quota} - \text{usage accumulator} - \text{rollover usage accumulator}) \leq 0$ For percentage intermediate limits, the limit is surpassed if $[(\text{usage limit} + \text{rollover quota}) \times \text{percentage intermediate limit} - \text{usage accumulator} - \text{rollover usage accumulator}] \leq 0$ For integer intermediate limits, the limit is surpassed if $(\text{intermediate limit} - \text{usage accumulator}) \leq 0$ <p>total is used to indicate limits total traffic.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none"> <code>ulVolume</code> for volume uplink <code>dlVolume</code> for volume downlink <code>bidirVolume</code> for volume bidirectional <code>time</code> for time



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].remaining["type"][n]₍₁₎₍₃₎</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	<p>The remaining usage of the selected Reporting Group for the subscriber or shared dataplan.</p> <p>The calculation of the value is $\text{usage limit} - \text{usage accumulator}$, if it is >0. Otherwise it is 0.</p> <p>Calculation of limits when Quota Rollover is enabled:</p> <ul style="list-style-type: none">• For absolute limits the value is calculated as $\text{usage limit} + \text{rollover quota} - \text{usage accumulator} - \text{rollover usage accumulator}$, if it is >0. Otherwise it is 0.• For percentage intermediate limits, the value is calculated as $(\text{usage limit} + \text{rollover quota}) \times \text{percentage intermediate limit} - \text{usage accumulator} - \text{rollover usage accumulator}$• For integer intermediate limits, the value is calculated as $\text{intermediate limit} - \text{usage accumulator}$ <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].currentRollover["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	<p>Value of the rollover usage accumulator of the selected Reporting Group for the subscriber or shared data plan.</p> <p>Default value is 0.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].currentRolloverPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].isRolloverSurpassed["type"]</code>	bool	true false	It indicates if the rollover usage limit of the selected Reporting Group for the subscriber or shared dataplan has been surpassed. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].rolloverFromPreviousPeriod["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover quota of the selected Reporting Group for the subscriber or shared dataplan is transferred from the previous period. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.receivedUsage.reportingGroup["total"/"reportingGroupName"].usageType["type"]⁽¹⁾</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.)	Usage reporting received for the subscriber for the indicated Reporting Group. ⁽⁴⁾ Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	Value of the usage accumulator of the selected Reporting Group for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].currentPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.



Table 4 Fair Usage Related Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].isLimitSurpassed["type"][n]⁽³⁾</code>	bool	true false	<p>It indicates if the usage limit of the selected Reporting Group for the subscriber or shared dataplan, and for current IP session has been surpassed.</p> <p>The limit is surpassed if:</p> $(\text{usage limit} - \text{usage accumulator}) \leq 0$ <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].remaining["type"][n]</code>	Integer	<p>Kilobytes (volume) or minutes (time) as a whole number.</p> <p>Precision of bytes and seconds is lost.</p>	<p>The remaining usage of the selected Reporting Group for the subscriber or shared dataplan, and current IP session.</p> <p>The calculation of the value is:</p> $\text{usage limit} - \text{usage accumulator}, \text{ if it is } > 0.$ <p>Otherwise it is 0.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time

(1) If aggregable dataplans are configured and an aggregated Reporting Group is selected, the value of this tag points to the aggregated usage accumulator.

(2) This tag always returns false for Postpaid subscriptions.

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

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

Table 5 Fair Usage Related Tags, Multiple Usage Accumulators

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



Table 5 Fair Usage Related Tags, Multiple Usage Accumulators

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].expiryDate["type"]</code>	string	Format: dd-mm-yyyyThh: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"> • volume for volume expiration • time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].isLimitSurpassed["type"][n]⁽¹⁾</code>	bool	true false	It indicates if the usage limit of the selected Reporting Group counter for the subscriber or shared dataplan has been surpassed. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].remaining["type"][n]⁽¹⁾</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	The remaining usage of the selected Reporting Group counter for the subscriber or shared dataplan. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].currentRollover["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover usage accumulator of the selected Reporting Group counter for the subscriber or shared dataplan. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].currentRolloverPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.



Table 5 Fair Usage Related Tags, Multiple Usage Accumulators

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].isRolloverSurpassed["type"]</code>	bool	true false	<p>It indicates if the rollover usage limit of the selected Reporting Group counter for the subscriber or shared dataplan has been surpassed.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].rolloverFromPreviousPeriod["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	<p>Value of the rollover quota of the selected Reporting Group counter for the subscriber or shared dataplan transferred from the previous period.</p> <p>Default value is 0.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	<p>Value of the usage accumulator of the selected Reporting Group counter for the subscriber or shared dataplan and current IP session.</p> <p>Possible values for type are as follows:</p> <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].currentPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.



Table 5 Fair Usage Related Tags, Multiple Usage Accumulators

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].counter["counterName"].isLimitSurpassed["type"][n]⁽¹⁾</code>	bool	true false	It indicates if the usage limit of the Reporting Group counter for the subscriber or shared dataplan and current IP address has been surpassed. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].remaining["type"][n]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	The remaining usage of the selected Reporting Group counter for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time

(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
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	Value of the usage accumulator of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan. ⁽¹⁾ When Quota Rollover is enabled, the calculation of the value is: usage accumulator + rollover usage accumulator Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].currentPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit. When Quota Rollover is enabled, the calculation of the value is: (usage accumulator + rollover usage accumulator) ÷ (usage limit + rollover quota) × 100



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"].expiryDate["type"]</code>	string	Format: dd-mm-yyyyThh:mm:ss	Date and time when the Reporting Group complementary usage accumulator with the specified groupName is reset for the subscriber or shared dataplan. ⁽¹⁾ Possible values for type are as follows: <ul style="list-style-type: none"> • volume for volume expiration • time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].remaining["type"][n]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	The remaining usage of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].currentRollover["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover usage accumulator of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].currentRollOverPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].isRolloverSurpassed["type"]</code>	bool	true false	It indicates if the rollover usage limit of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan has been surpassed. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time



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"].rolloverFromPreviousPeriod["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover quota of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan, transferred from the previous period. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].current["type"]⁽⁴⁾</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	Value of the usage accumulator of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].currentPercentage["type"]⁽⁵⁾</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].expiryDate["type"]⁽³⁾</code>	String	Format: dd-mm-yyyyThh:mm:ss	Date and time when the absolute accumulated usage for the Reporting Group with the specified groupName expires for the subscriber or shared dataplan. Possible values for type are as follows: <ul style="list-style-type: none"> • volume for volume expiration • time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].hasExpired["type"]⁽⁵⁾</code>	String	true false ⁽⁵⁾	It returns true after expiration date, if expiration date is before the endDate, and false otherwise. Possible values for type are as follows: <ul style="list-style-type: none"> • volume for volume expiration • time for time expiration
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].remaining["type"][n]⁽²⁾⁽³⁾</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	The remaining usage of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time



Table 6 Fair Usage Related Tags, Multiple Service Offerings

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].selected³⁾</code>	bool	true false	Value of the rollover usage accumulator of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan. It returns true for the active subscriber group, and false otherwise.
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].currentRollover["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover usage accumulator of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • u1Volume for volume uplink • d1Volume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].currentRolloverPercentage["type"]</code>	Integer	0-100	Same as above, but value is expressed as percentage of the corresponding limit.
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].isRolloverSurpassed["type"]</code>	bool	true false	It indicates if the rollover usage limit of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan has been surpassed. Possible values for type are as follows: <ul style="list-style-type: none"> • u1Volume for volume uplink • d1Volume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].rolloverFromPreviousPeriod["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number.	Value of the rollover quota of the selected Reporting Group with the specified groupName for the subscriber or shared dataplan is transferred from previous period. Default value is 0. Possible values for type are as follows: <ul style="list-style-type: none"> • u1Volume for volume uplink • d1Volume for volume downlink • bidirVolume for volume bidirectional • time for time



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"].counter["counterName"].current["type"]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	Value of the usage accumulator of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.session.accumulatedUsage.reportingGroup["total"/"reportingGroupName"].group["groupName"].counter["counterName"].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"].counter["counterName"].remaining["type"][n]</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes or seconds is lost.	The remaining usage of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time
<code>AccessData.subscriber.session.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 usage accumulator of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none"> • ulVolume for volume uplink • dlVolume for volume downlink • bidirVolume for volume bidirectional • time for time



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]⁽²⁾</code>	Integer	Kilobytes (volume) or minutes (time) as a whole number. Precision of bytes and seconds is lost.	The remaining usage of the selected Reporting Group counter with the specified groupName for the subscriber or shared dataplan and current IP session. Possible values for type are as follows: <ul style="list-style-type: none">• ulVolume for volume uplink• dlVolume for volume downlink• bidirVolume for volume bidirectional• time for time

(1) The group name is the dataplan.

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

(3) If aggregable dataplans are configured and the `_Aggregated_` group name is used, the value of this tag points to the aggregated usage accumulator.

(4) The group name is either the dataplan, or `_Aggregated_` in case of aggregated usage accumulator.

(5) This tag always returns false for Postpaid subscriptions

The following tags can be used only for Stackable Dataplans. If the specified group is not associated with multiple instances, the SAPC does not evaluate the policy rules.

Table 7 Multiple Instances Tags

Tag	Return Type	Possible Values	Comments
<code>AccessData.subscriber.accumulatedUsage.group["groupName"].currentInstance</code>	string	Format: Start Date "dd-mm-yyy yThh:mm:ss", Stop Date "dd:mm-yyyyThh:mm:ss", Index "n"	Duration and index of the instance being used for the absolute usage accumulator of the prepaid group (with the specified groupName). If the instance is not overlapping with others, the value of index is 1; otherwise, the value is the number of instances activated in time order.



Table 7 Multiple Instances Tags

Tag	Return Type	Possible Values	Comments
AccessData.subscriber.accumulatedUsage.group["groupName"].isInstanceAvailable	bool	true false	It indicates whether any instance is available for the absolute usage accumulator of the prepaid group (with the specified groupName). It returns true: <ul style="list-style-type: none"> when the instance in use is still valid any new instance can be used in future
AccessData.subscriber.accumulatedUsage.group["groupName"].newInstanceStart	bool	true false	It indicates whether a new instance starts to be used for the absolute usage accumulator of the prepaid group (with the specified groupName).

16.1 Fair Usage Policy Tags in Dynamic Group Selection Policies

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

Note: The aggregated limits data can be used in the policy conditions. But the Dynamic Group Selection policies are not allowed for the subscriber groups associated with aggregable Reporting Groups. Otherwise, the policy results could be unexpected.

Table 8 Fair Usage Policy Tags in Dynamic Group Selection Policies

Tag	Comments
<...>.group["groupName"].<...>.remaining["type"][n] ⁽¹⁾ <...>.group["groupName"].<...>.current["type"][n]	They should be used, including group name definition, which is the dataplan or <code>_Aggregated_</code> in case of aggregated usage accumulator, when policies are required to detect when the limit is surpassed in cases of having the same Reporting Group associated to different groups.
<...>.isLimitSurpassed["type"][n] ⁽²⁾	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.
<...>.group["groupName"].<...>.hasExpired["type"][n] ⁽¹⁾	It should be used, including group name definition, which is the dataplan or <code>_Aggregated_</code> in case of aggregated usage accumulator, when policies are required to detect an expired Prepaid Reporting Group in cases of having the same Reporting Group associated to different groups.

(1) If aggregable dataplans are configured and the `_aggregated_` group name is used, the value of this tag points to the aggregated usage accumulator.

(2) If aggregable dataplans are configured and an aggregated Reporting Group is selected, the value of this tag points to the aggregated usage accumulator.