Configuration Guide for ADC based on PCC rules (Gx)

Ericsson Service-Aware Policy Controller

USER GUIDE

32/1553-CSH 109 215/7 Uen B

Abstract

This document is a guideline to configure the SAPC node to support Application Detection and Control (ADC) for PCC rules over Gx interface.

Contents

1 Introduction

1.1 Document Purpose and Scope

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

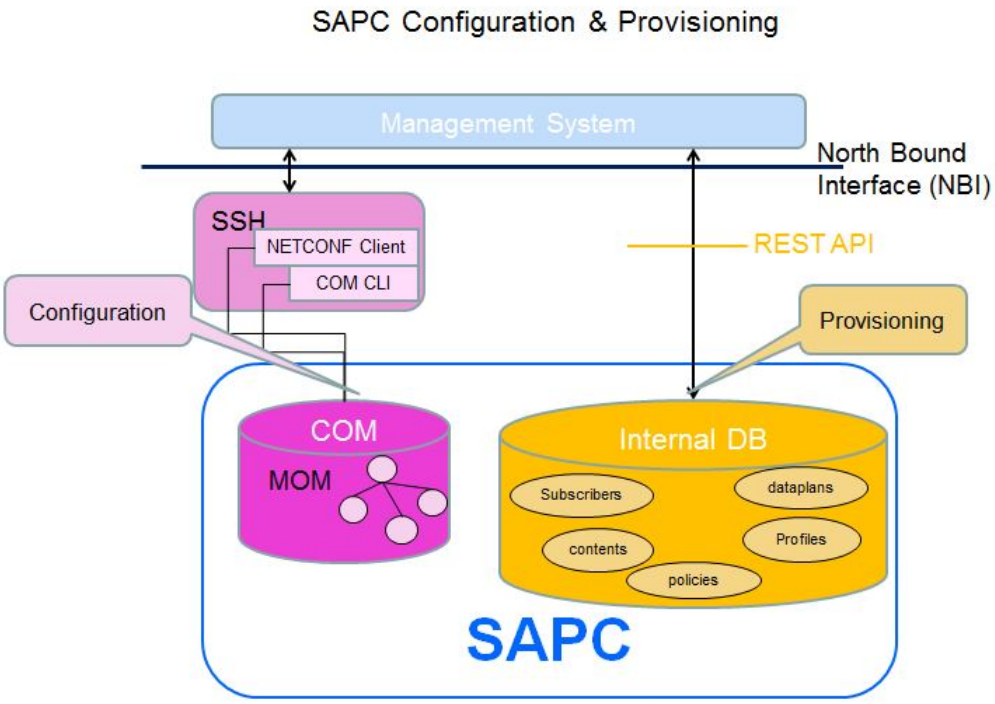

Figure 1 Configuration and Provisioning Overview

The purpose of this document is to provide guidelines to configure the SAPC node for Application Detection and Control (ADC) by providing configuration examples.

The configuration for ADC is an extension to Service Access Control, covered in Configuration Guide for Access and Charging Control (Gx).

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

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

1.2 Typographic Conventions

This document uses the following typographic conventions:

Table 1 Typographic Conventions

Convention	Description	Example
MOC	COM Model Object Class	`DiameterNode`

NETCONF	SAPC COM configuration	`<edit-config>`
		`<target>`
		`<running/>`
		`</target>`
		`<config>`
		`<ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">`
		`<managedElementId>1</managedElementId>`
		`<PolicyControlFunction xmlns="urn:com:ericsson:ecim:sapcmom`
		`<policyControlFunctionId>1</policyControlFunctionId>`
		`<Network xmlns="urn:com:ericsson:ecim:networkmom">`
		`<networkId>1</networkId>`
		`<DiameterNodes>`
		`<diameterNodesId>1</diameterNodesId>`
		`<diameterNodeId>ggsnNodeHostname.ggsnNodeHostRe`
		`<controls>IP_CAN_SESSION_ACCESS</controls>`
		`<dynamicServiceSupport>true</dynamicService`
		`<adcSupport>true</adcSupport>`
		`</DiameterNode>`
		`</DiameterNodes>`
		`</Network>`
		`</PolicyControlFunction>`
		`</ManagedElement>`
		`</config>`
		`</edit-config>`

REST	SAPC provisioning.
	
	Exact URI in the provisioning REST API names, fields, or their corresponding values.
		`PUT /contents/Skype`
		`{`
		`"contentName" : "Skype","pccRuleName" : "8001", "pccRuleType" :`
		`}`

1.3 Other Conventions

This document refers to some configuration and provisioning data.

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

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

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

  Example: set `enableReauthsOnSubsChange` attribute in class AppConfig.

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

  a NETCONF interface, refer to Ericsson NETCONF Interface.

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

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

  b Or COM CLI, refer to Ericsson Command-Line Interface.

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

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

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

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

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

  where /resource-URI is the relative URI from the SAPC provisioning base URI detailed in Provisioning REST API.

  Example:

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

Note: To ease provisioning operations, the SAPC provides an HTTPS CLI client named `resty`, refer to Provisioning Tools.

2 Configuration Prerequisites

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

— CBA Components are installed.

— The SAPC product software is installed.

— To have a detailed understanding of the function.

3 Configuration

3.1 Configuring Gx PCEFs

To support ADC for some PCEFs, set the `adcSupport` attribute to `true` at the `DiameterNode` level.

The SAPC downloads preconfigured PCC rules enhanced with ADC only to the PCEF with both `adcSupport` and `dynamicServiceSupport` attributes set to `true`. Static PCC rules enhanced with ADC are activated in all the PCEFs with `adcSupport` attribute set to `true`.

Note: In multiple Gx scenarios where an IP-CAN session is controlled by several PCEFs, ADC support can be configured for more than one PCEF. But each PCEF may report application status and perform enforcement independently.

3.2 Provision ADC for Services

To indicate the application identifier for application detection to the PCEF, provision the value of the `tdfAppId` attribute using the `content` URI in the provisioning REST API.

Note: For static PCC rules, the `tdfAppId` attribute is optional. Even if it is config ured, the SAPC does not send the `TDF-Application-Identifier` AVP to the PCEF.

Example:

```
PUT /contents/Skype
{
   "contentName" : "Skype",
   "pccRuleName" : "8001",
   "pccRuleType" : 2,
   "tdfAppId" : "appSkype"
}
```

Note: Same `tdfAppId` can be provisioned in different services, but if this is the case, take care that only one PCC rule with same `TDF-Application-Identifier` should be active at any time.

3.3 Configure ADC Redirection

ADC Redirection is only applicable to preconfigured PCC rules enhanced with ADC.

Provision the ADC Redirection profiles using `content-adc-redirect` URI in the provisioning REST API.

Example:

```
PUT /profiles/content-adc-redirect/AdcRedirectProfile1
{
   "profileId" : "AdcRedirectProfile1",
   "addressType" : "IPv6_ADDRESS",
   "address": "4055:0db8:85a3::1319:8a2e:0370:1111"
}
```

3.3.1 Configure ADC Redirection Unconditionally

To assign the ADC Redirection profile unconditionally to a service, use the `contentAdcRedirectProfileId` inside `staticQualification` JSON attribute of the `content` URI in the provisioning REST API.

Example 1 ADC Redirection Using Unconditional Configuration

3.3.2 Configure ADC Redirection Conditionally

Using policies, it is possible to use dynamic conditions to assign ADC Redirection to services. This can be done in addition to unconditional service qualification explained in Section 3.3.1 on page 8 . The policy types related to ADC Redirection which can be used and configured in the SAPC are shown in Figure 2:

Application Detection and Control (ADC) Redirect Policies

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
ADC Redirect	adc-redirect	<contentId>	<subscriberId> <dataplanId>	permit **adc-redirect** AdcRedirectProfile ["<redirectProfileName>"] permit **adc-redirect** "doNotRedirect"	Type II = Mixing policies and qualification **Gx** Conditions: Access Data Subscriber ToD Used to return Redirect-Information AVP in Gx PCC Rules

Figure 2 Application Detection and Control Redirection Policies

Note: The values configured using policies prevail over the unconditional profiles provisioned in the service. The SAPC considers qualification data provisioned unconditionally, only if no policy for ADC redirection is applicable to a service or if the policies do not return any result.

Example 2 ADC Redirection Using Dynamic Conditions

3.4 Configure ADC Mute Notification

ADC Mute Notification is only applicable to preconfigured PCC rules enhanced with ADC.

Note: Once The SAPC sets a Mute Notification value, it is persisted until the service is removed.

3.4.1 Configure ADC Mute Notification Unconditionally

To indicate whether to mute notifications for a specific application associated with preconfigured PCC rules enhanced with ADC, use the `contentAdcMuteNotification` attribute inside `staticQualification` JSON attribute of the `content` URI in the provisioning REST API:

— `muted`

— `unmuted` (default value)

Example:

```
PUT /contents/WeChat
{
   "contentName" : "WeChat",
   "pccRuleName" : "9001",
   "pccRuleType" : 2,
   "tdfAppId" : "appWeChat",
   "staticQualification":
   {
      "contentAdcMuteNotification": "\"muted\""
   }
}
```

3.4.2 Configure ADC Mute Notification Conditionally

Using policies, it is possible to use dynamic conditions to assign Mute Notifications (applicable during the whole lifetime of the PCC rules). This can be done in addition to unconditional service qualification explained in Section 3.4.1 on page 10 . The policy types related to ADC which can be used and configured in the SAPC are shown in Figure 3:

Application Detection and Control (ADC) Mute Notification Policies

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
ADC Mute	adc-mute-notification	<contentId>	<subscriberId> <dataplanId>	permit **adc-mute-notification** "muted" or "unmuted"	Type II = Mixing policies and qualification **Gx** Conditions: Access Data Subscriber Time conditions (but no ToD reauthorization) Used to return Mute-Notification AVP in Gx PCC Rules

Figure 3 Application Detection and Control Mute Notification Policies

Note: If time of day conditions are used in the policies, the SAPC does not perform time-based reauthorization for Mute Notifications.

The values configured using policies prevail over the Mute Notification value provisioned unconditionally in the service. The SAPC considers Mute Notification value provisioned unconditionally, only if no policy for Mute Notification is applicable to a service or if the policies do not return any result.

Example 3 ADC Mute Notification

3.5 Configure Event Triggers

To indicate to the PCEF that the SAPC is interested in application reporting, use the values `APPLICATION_START` and `APPLICATION_STOP`.

To know how to configure `EventTriggers`, refer to Configuration Guide for Access and Charging Control (Gx).

4 Configuration Examples for Use Cases

4.1 QoS Control based on Application Detection

The following example shows how Application Detection can be used to upgrade QoS for a dedicated bearer, when specific application traffic is detected by the PCEF and reported to the SAPC.

Example 4 QoS Upgrade

5 Appendix A. ADC Policy Types

Figure 4 shows the policy types related to ADC, which can be used and configured in the SAPC.

Application Detection and Control (ADC) Policies

Policy Type	Policy Locator			Output Attributes	Comments
	Context	Resource	Subject		
ADC Redirect	adc-redirect	<contentId>	<subscriberId> <dataplanId>	permit **adc-redirect** AdcRedirectProfile ["<redirectProfileName>"] permit **adc-redirect** "doNotRedirect"	Type II = Mixing policies and qualification **Gx** Conditions: Access Data Subscriber ToD Used to return Redirect-Information AVP in Gx PCC Rules
ADC Mute	adc-mute-notification	<contentId>	<subscriberId> <dataplanId>	permit **adc-mute-notification** "muted" or "unmuted"	Type II = Mixing policies and qualification **Gx** Conditions: Access Data Subscriber Time conditions (but no ToD reauthorization) Used to return Mute-Notification AVP in Gx PCC Rules

Figure 4 Application Detection and Control Policies

6 Appendix B. ADC Policy Tags

The following ADC policy tag can be used in the condition formula of rules.

Table 2 ADC Policy Tag

Tag	Return Type	Possible Values	Comments
`AccessData.tdfApp["id"].isStarted`(1)	Boolean	true false	The current state for the TDF application: • True –There are no instances (reporting done at TDF-Application-Id level): when the last event received by the SAPC was a start. –There are multiple instances: when for any of the instances under the TDF-Application-Id, the last event received by the SAPC was a start. • False –Before receiving any event from the PCEF. –There are no instances (reporting done at TDF-Application-Id level): when the last application event received by the SAPC was a stop. –There are multiple instances: when for every instance under the TDF-Application-Id, the last event received by the SAPC was a stop.

(1) The use of this tag does not make sense for some policy types such as Autoprovisioning or Dynamic OCS selection.

Note: When a PCC rule enhanced with ADC is not authorized, the latest application status may not be reported by the PCEF. In this case, the value of this tag is not up to date.

Glossary

ADC
Application Detection and Control

API
Application Programming Interface

CBA
Component Based Architecture

HTTP
Hypertext Transfer Protocol

JSON
JavaScript Object Notation

PCC
Policy Charging Control

PCEF
Policy Charging Enforcement Function

PCRF
Policy and Charging Rule Function

QoS
Quality of Service

REST
Representational State Transfer

SAPC
Ericsson Service-Aware Policy Controller

TDF
Traffic Detection Function

Reference List

Ericsson Documents

[1] Configuration Guide for Access and Charging Control (Gx)

[2] Managed Object Model (MOM)

[3] Provisioning REST API

[4] System Administrator Guide