



Cisco Broadband IP Services Module Application Guide

Software Release 3.0

July 15, 2005

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: OL-7768-01
Text Part Number: OL-7768-01



THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

o Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, ce marks of Cisco Systems, Inc.; and Access Registrar, Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Ci Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Empowering the In erChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expert rd, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Ro X, ScriptShare, SlideCast, SMARTnet, StrataView Plus, TeleRouter, The Fastest Way to Increase Your Internet Quotient, and TransPath ystems, Inc. and/or its affiliates in the United States and certain other countries.

mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partner y other company. (0502R)

Cisco Broadband Services Modules Application Guide
© 2005 Cisco Systems, Inc. All rights reserved.

Overview	1
Audience	1
Organization	1
Conventions	1
Text	2
Icons	3
BPM Documentation Set	3
Broadband Policy Design Studio User Guide	3
Broadband Policy Manager Operations Guide	3
Broadband Policy Manager Release Notes	3
Broadband Policy Manager Software Installation Guide	3

CHAPTER 1

Introduction	1
Overview	1
Broadband IP Services Modules Applications	1
Bandwidth-on-Demand Application	2
Time-Based Turbo Bandwidth Application	2
Tiered Applications	2
Quota-Based Tiered Applications	2
VoIP-Oriented Preferential Applications	3
Firewall Application	3
Network Domain Components	3
Retail Network	4
SP Director BPM	4
SP Policy Repository BPM	5
Datacaster	5
Initial Data Load	7
Return to Initial State	8
Failure Recovery	8
Quotas	8
Supplements	8
Service Bundles and Packages	8
Initial Login Bandwidth	9
Allowed Subscriber Services	9
Boost Package Support	9
Maximum Usage Values	9
Action Taken Upon Volume Quota Breach	10
Quota Triggers	10
Supplemental Volume Limits	10

- Quota Resets 10
- Statistics Tracking 10
- Usage Statistics 10
- SP Session Controller BPM 10
 - Login 11
 - Profile 11
- Accounting Usage Updates 12
- Breach Notifications 12
- Volume Supplement 13
- Reset Subscriber Quotas/Monthly Rollover 13
- Administrator Services 13
- Wholesale Network 13
 - NP Director BPM 14
 - NP Policy Repository BPM 14
 - NP Session Controller BPM 14
 - .10K BRAS Device 14
 - RADIUS Server 14
- Secure Access Model 16
- Broadband Policy Design Studio 17
 - Flow 18
 - Service 19
 - Rule 19
 - Policy Function 19
 - Agent 20
 - Device Types 20
 - Device Rule 20
 - Device Profile 21

CHAPTER 1

- Configuration 1**
 - Overview 1
 - Retail Domain Devices 2
 - Director BPM 2
 - Procedure: Configure the SP Director BPM 3
 - Policy Repository BPM 4
 - Procedure: Configure the SP Policy Repository BPM 4
 - Session Controller BPM 5
 - Procedure: Configure the SP Session Controller BPM 5
 - Procedure: Configure Session Controller Agents 6
 - Wholesale Domain Devices 7

Policy Repository BPM 8
 Procedure - Configure the NP Policy Repository BPM 8
 Session Controller BPM 9
 Procedure: Configure the NP Session Controller BPM 9

CHAPTER 1

System Specific Services 1

Overview 1
 SP Director BPM 1
 NP Director BPM 2
 NP Session Controller BPM 2
 SP Session Controller BPM 2
 NP Policy Repository BPM 3
 SP Policy Repository BPM 3

CHAPTER 1

Deployment 1

Overview 1
 Architecture 1
 Nonregulated Deployment 1
 Regulated Deployment 2
 NP SP Deployment Detail 3
 Failover 3
 Data Structure and Repository 3
 System Support Type 4
 Network Support 4
 Service Support 4
 Combined Support 4
 Service or Network Director Support 4
 Setting System Type 4
 Metadata 5
 WSDL Files 5
 Common Architecture 5
 Service Access 5
 Service Event 6
 Service Event Quota 6
 Service Profile 6
 Flow 7
 Service Profile Quota 7
 Quota Supplement 7

- Usage Threshold 7
- Administrator 8
- BPM Role Definition 8
 - Network 8
 - Service 8
 - Combined 8
- Remote Data Store 9
 - Data Type 9
 - Session and Policy Database on Local BPM 9
 - Session Database on Remote BPM / Policy Database on Local BPM 10
 - Policy Database on Remote BPM / Session Database on Remote BPM 10
 - Session and Policy Database on Remote BPM 10
 - Remote Database Store Package Requirements 10
- Timeout 10
- Rule and Policy Function 11
 - Rule 11
 - Policy Function 11
- Network Configuration 11
 - Device 11
 - Device Type 12
 - Device Rule 12
 - Device Script 12
 - Allowed Value 12
 - Management Protocol 12
 - Profile 13
 - Network Event 13
 - Profile Proxy 13
 - Access Line 13
 - Access Group 13
- Service Configuration 14
 - Domain 14
 - Service Group 14
 - Subscriber 14
 - Registered Application 15
- Director Configuration 15
 - SP Director Entry 15
 - NP Director Entry 15
 - NAS Director Entry 15

Applications in Action	1
Overview	1
BPM and 10K Applications	1
Bandwidth on Demand	2
How It Works	2
BPM Functions	3
Access Methods	3
BPM Agents	3
Base Capabilities	4
Application Benefits	4
Time-Based Turbo Bandwidth	5
How It Works	6
BPM Functions	6
Access Methods	7
BPM Agents	7
Base Capabilities	7
Application Benefits	7
BPM, 10K, and SCE 1010	8
Tiered Applications	8
How It Works	9
BPM Functions	9
Access Methods	9
BPM Agents	10
Base Capabilities	10
Application Benefits	10
Quota-Based Tiered Applications	11
How It Works	11
BPM Functions	12
Access Methods	12
BPM Agents	12
Base Capabilities	12
Application Benefits	13
VoIP-Oriented Preferential Applications	14
How It Works	14
BPM Functions	15
Access Methods	15
BPM Agents	15
Base Capabilities	15
Application Benefits	15

Firewall Application 16
 How It Works 17
 BPM Functions 17
 Access Methods 17
 BPM Agents 17
 Base Capabilities 17
 Application Benefits 18

CHAPTER 1

Application Subscriber Login and RADIUS Processing 1

Overview 1
 RADIUS Accounting Message Interception 1
 Service - *genericRadiusStartAccounting* 2
 Service - *genericRadiusStopAccounting* 2
 Service - *genericNetworkStateMachine* 3
 Service - *deviceConfig* 3
 Service - *deviceConfigISG* 3
 RADIUS Accounting Proxy/Accounting Server Start Accounting Processing 4
 RADIUS Accounting Proxy/Accounting Server Stop Accounting Processing 5
 RADIUS Accounting Proxy/Accounting Server Accounting On/Off Processing 5
 Application Network State Machine 5
 Event - *ntwkStartSession* 6
 Network Session Created 6
 Invalid State 7
 Event - *ntwkInstallDefaultProfile* 7
 Network Install Complete 7
 Invalid State 8
 Event - *ntwkEndSession* 8
 Network Install Complete and Active 8
 Logoff In Progress 9
 Invalid State 9
 SP Session Controller Message Processing and State Machine Support 9
 Event - *serviceSessionBegin* 9
 Event - *serviceSessionDefaultProfileInstalled* 10
 Event - *serviceSessionEnd* 11
 Inter BPM Communication 11
 NP Session Controller BPM 12
 Service - *subscriberNetworkToServiceRequest* 12
 Service - *subscriberServiceToNetworkEvent* 12
 Subscriber-based Events from the NP Session Controller BPM to the SP BPM 12

SP Session Controller BPM	13
Service - <i>subscriberServiceToNetworkRequest</i>	13
Service - <i>subscriberServiceProcessNetworkEvent</i>	13
Subscriber-based events from the SP Session Controller BPM to the NP Session Controller BPM	13
NP Director BPM	13
Service - <i>subscriberNetworkToServiceEvent</i>	13
Service - <i>subscriberServiceToNetworkEvent</i>	14
Service - <i>deliverSubscriberNetworkEvent</i>	14
Service - <i>deliverSubscriberServiceEvent</i>	14

CHAPTER 1

External Application Interface Communications	1
Overview	1
Session Controller Registration with SP Director BPM	1
External Application Registration	1
Service - <i>registerApplication</i>	2
Service - <i>unregisterApplication</i>	2
Utility Functions	3
Function - <i>getSubscriberExternalIndex</i>	3
Profile Processing Service	3
Function - <i>applyProfile</i>	3
Function - <i>releaseProfile</i>	4
Feature Processing Service	5
API - <i>applyProfileFeature</i>	5
Supported Features and Services Data Values	6
Service - <i>changeSubscriberServiceLevel</i>	6
Service - <i>changeSubscriberServiceGroup</i>	7
Service - <i>supplementQuota</i>	7
Inter BPM Communication	7
SP Session Controller BPM	7
Service - <i>subscriberProcessProfileEvent</i>	7
Service - <i>sendPCSRegistration</i>	8
Service - <i>sendSubscriberIPAddress</i>	9
Service - <i>sendCacheRequest</i>	9
SP Policy Repository BPM Interface	10
Service - <i>processPCSRegistration</i>	10
Service - <i>updateSubscriberIPAddress</i>	11
Service - <i>processCacheRequest</i>	11
Service Director BPM	11

[Service - *applyProfile*](#) 11
[Service - *releaseProfile*](#) 12
[Service - *applyProfileFeature*](#) 12
[Service - *getSubscriberExternalIndex*](#) 13
[Service - *registerApplication*](#) 13
[Service - *unregisterApplication*](#) 14
 SP Session Control Interface 14
[Service - *subscriberProcessProfileEvent*](#) 14

CHAPTER 1

Agents 1
 Overview 1
 Application Agents 1
 Event 1
 Netconf 2
 PCube 2
 RADIUS 2
 Scheduler 2
 Socket 3
 SOAP 3
 TCL 3
 Device Agents 3
 SP Repository 3
 SP Director 4
 NP Repository 4

CHAPTER 1

Platform Specifications 1
 Overview 1

Preface

Overview

This document discusses the Cisco Broadband IP Services Module (BSM) offering that runs on the Broadband Policy Manager (BPM) system. It discusses the BSM applications, their position in the BPM architecture, BPM functionality, and the Broadband Policy Studio (BPS) and Broadband Policy Design Studio (BPDS) the graphical user interfaces. The document gives an overview of the BPM system and BSM components and explains the benefits they provide for service providers and their customers. It also discusses deployment, prebuilt tools, system specific services, and supported platforms.

Before you use the information in this document, complete the installation of the hardware and software that came with your BPM system according to the instructions in the documentation.

Audience

This guide is for the network professional who handles the BSM capability for wholesale and retail domains.

Organization

This document contains ten chapters:

- Chapter 1 - [Introduction](#)
- Chapter 2 - [Configuration](#)
- Chapter 3 - [System Specific Services](#)
- Chapter 4 - [Deployment](#)
- Chapter 5 - [Applications in Action](#)
- Chapter 6 - [Application Subscriber Login and RADIUS Processing](#)
- Chapter 7 - [External Application Interface Communications](#)
- Chapter 8 - [Agents](#)
- Chapter 9 - [Platform Specifications](#)

The document also contains one appendix:

- Appendix A - [Glossary](#)

Conventions

This guide uses the text and icon conventions described in this section.

Text

The table below contains documentation text conventions.

Convention	Explanation	Example
alternate mouse button	Usually indicates the right mouse button.	Select the agent with the alternate mouse button.
arrow -->	Indicates the selection order of menu items.	File --> Save Go to the File menu and select the Save function.
bar brackets []	Indicate the default.	Select your Name Service type [2]: This indicates the default is 2.
bold	Indicates user input or button selection.	poweron
<i>bold italic</i>	Indicates objects, attributes, pin names, and service flows.	On the backend, monitor the policy database session table and verify that session exists and <i>sessionstate</i> is <i>Accounting Started</i> .
Ctrl+X	Indicates the quick access key for a menu option.	Ctrl+M This indicates open the Object Manager.
default mouse button	Usually indicates the left mouse button.	Select the agent with the default mouse button.
<i>italic</i>	Indicates an application, chapter, directory, document, header, section, or title names.	For more information, refer to the section entitled <i>Creating Services</i> .
<KEYNAME>	Indicates press the named key.	Supply the required information, then press the <ENTER> key.
screen display	Represents system output.	This agent does not have agent-specific properties.

Icons

This guide may use the following icon conventions to provide additional information and to indicate special conditions or possible risks:



Note

A note is an informational message containing a tip or suggestion.



Caution

A caution indicates a risk of damage to equipment or a loss of data.



Warning

A warning alerts you to the risk of personal injury.

BPM Documentation Set

In addition to this book, the documentation for your BPM system includes the following documents:

- Broadband Policy Design Studio User Guide
- Broadband Policy Manager Operations Guide
- Broadband Policy Manager Release Notes
- Broadband Policy Manager Software Installation Guide

Broadband Policy Design Studio User Guide

This guide provides instructions for installing the BPDS, and using it to create, deploy, and manage network services, and to manage subscribers.

Broadband Policy Manager Operations Guide

This guide describes the use of the BPDS to obtain information, conduct day-to-day operations, perform maintenance tasks, and troubleshoot problems with the BPM system.

Broadband Policy Manager Release Notes

This document describes new features, known limitations, and other important information about the BPM system.

Broadband Policy Manager Software Installation Guide

This guide describes the installation of the BPM software for the BPM.

Introduction

Overview

The Broadband IP Services Module (BSM) offering provides a series of applications, to be deployed by service providers (SPs), and a baseline architecture for service providers to build additional dynamic, subscriber-based, IP services. The BSM module runs on the Broadband Policy Manager (BPM) deployment architecture. The BPM system includes the applications and prebuilt components. It also includes the Broadband Policy Design Studio (BPDS), the graphical user interface that the service provider can use to develop and manage BPM components.

This chapter discusses the following topics:

- [Broadband IP Services Modules Applications](#)
- [Network Domain Components](#)
- [Secure Access Model](#)
- [Broadband Policy Design Studio](#)

Broadband IP Services Modules Applications

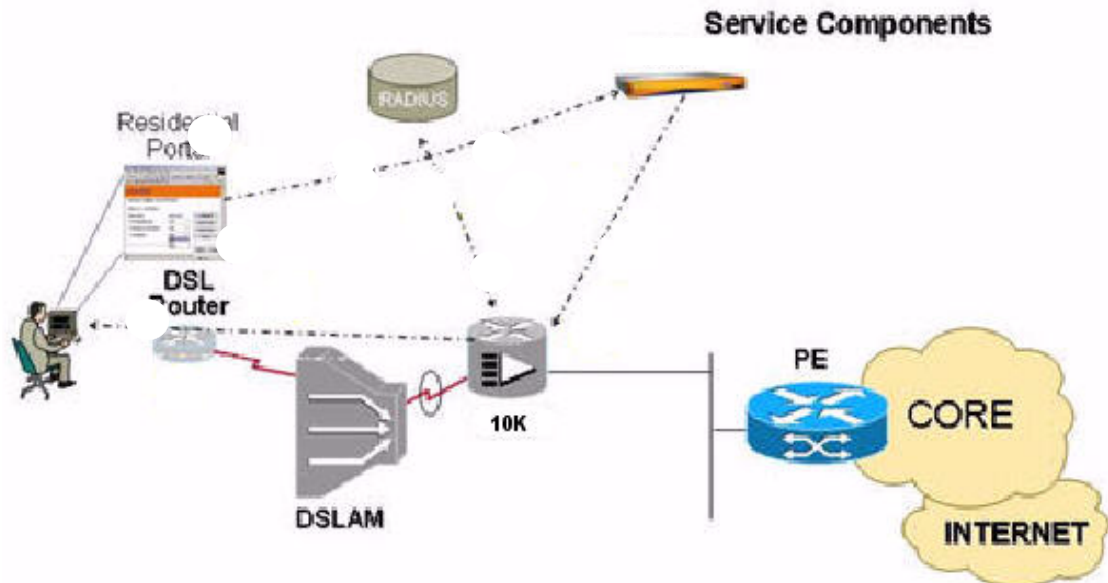
The BSM module offers the following applications:

- [Bandwidth-on-Demand Application](#)
- [Time-Based Turbo Bandwidth Application](#)
- [Tiered Applications](#)
- [Quota-Based Tiered Applications](#)
- [VoIP-Oriented Preferential Applications](#)
- [Firewall Application](#)

Bandwidth-on-Demand Application

With Bandwidth-on-Demand (BoD), the subscriber can dynamically change his Internet bandwidth, without intervention by the service provider. The service provider can provide additional revenue-producing services, beyond the basics. Figure 1-1 represents a Bandwidth-on-Demand application.

Figure 1-1 Sample BSM Application - Bandwidth-on-Demand.



Time-Based Turbo Bandwidth Application

The Time-Based Turbo Bandwidth application is driven by the subscriber contract. During slow system-usage times, the administrator can vary subscriber or supplier contract terms, so that spare capacity is managed wisely.

Tiered Applications

The Tiered Applications package lets the service provider bundle applications for specific subscriber bases. This package uses the familiar gold, silver, and bronze Internet service levels. It identifies a subscriber and automatically deploys his tiered application package. A subscriber with one tier package can modify his service by selecting an alternative package.

Quota-Based Tiered Applications

The Quota-Based Tiered Applications package lets the service provider bundle applications and apply quotas to any application within the service tier. If the subscriber breaches his quota, the SP BPM can deploy controls that restrict or diminish the application performance, or offer the subscriber the opportunity to purchase incremental quota.

VoIP-Oriented Preferential Applications

With a basic Internet subscriber service package, a subscriber receives standard telephone access. If a Voice-over-IP (VoIP) package is supported by his provider, the subscriber can opt to receive the *Internet with VoIP* package, which provides for calls with an improved service quality.

Firewall Application

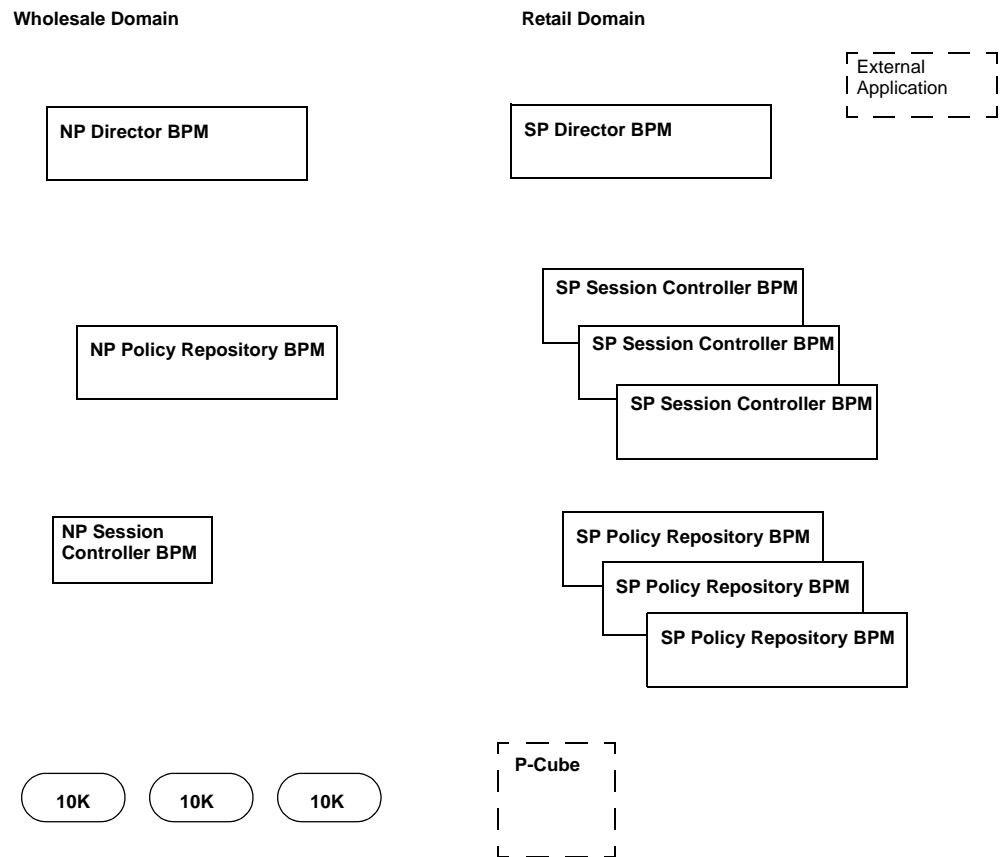
This application allows a subscriber to control access to services and web sites that can, and cannot, be accessed from his subscriber line. He can tailor the available services to govern when and what his subaccounts can access. A service provider can use this application to control access to applications and web sites and to limit service access for specific groups.

Network Domain Components

Figure 1-2 presents a view of BPM network components, divided into wholesale (NP) and retail (SP) domains. The components hold service provider policies and pass those policies to applications. Applications issue BPM service requests, based on the policies. BPM translates the policies into corresponding BPM rules to apply and wholesale BPM policies to invoke. The wholesale or retail domains in the figure can belong to two different companies. However, this figure depicts wholesale and retail networks as two groups within the network of one company. The depicted network contains the following BPM components:

- Director BPM
- Policy Repository BPM
- Session Controller BPM

Figure 1-2 Wholesale and Retail Domain Devices.



Retail Network

The retail network, on the right in Figure 1-1, represents the service provider that offers the BSM features to the subscriber. Retail network components manage subscriber policies, based on interactions with the application environment. As applications request the activation and deactivation of network policies, and deliver their related events, the BSM solution controls the underlying transport network to deliver the desired traffic contracts.

The following represent retail components:

- SP Director BPM
- SP Policy Repository BPM
- SP Session Controller BPM

SP Director BPM

The SP Director BPM handles the individual session requests from the Application layer. The director uses the Service Directory Entry table to route requests to the appropriate BPM devices. The Director BPM has a route table that correlates live subscriber sessions to specific Session Controller BPMs. This ensures applications do not know the physical binding between subscribers and the network.

Although the SP Director BPM receives requests from the subscriber via the Application layer, it does not perform actions on the requests. It consults its local database and locates the Session Controller BPM that handles the devices that are connected to the subscriber. It directs the subscriber request to the correct Session Controller BPM for processing.

The SP Director BPM allows scalability and hides the internal mappings of the solution from external entities such as the applications or the Application layer.

The SP Director BPM processes requests from the application environment using the Transaction Remote Procedure Call (TRPC) interface. The SP Director BPM distributes all service requests among the available Session Controller BPM clusters.

The SP Director BPM receives requests from the Application layer. It consults a local table containing a mapping of the subscriber identifier to the retail Session Controller BPM. The identifier determines the access line for application of a profile. On login, SP Session Controller BPMs update the table as network sessions start, ensuring that the SP Director BPM always has real-time mapping knowledge.

During calls to *ApplyPolicy*, the SP Director BPM retains the *sessionID* assigned by a successful execution.

SP Policy Repository BPM

In the retail network, the SP Policy Repository BPM handles database requests from NP Session Controller BPMs, NP Session Controller BPMs, and NP Director BPMs. As the central database for the Session Controller BPMs, the SP Policy Repository BPM component stores personal subscriber and configuration data. It holds the data required for normal operation, such as policies and rules, service package definitions, identifier mappings, usage statistics, and logs. Each Session Controller BPM has its own disk space and caches the data required for its operation to minimize calls to the Policy Repository.

When a subscriber is provisioned, the Access Line ID and an identifier are created on the Policy Repository BPM. The network uses this unique identifier, instead of a username, network, and password. Only the two endpoints of the network know the subscriber identity. In this way, his identity does not traverse the network.

Datacaster

A Datacaster process runs on every BPM device and may be both a subscriber and a publisher. In addition to the Datacaster agent interface, the Datacaster process also supports a command line interface (CLI). Data moves between datacasters, rather than by direct database connections, except for the initial set copy.

The Datacaster tool moves data from one place to another. Datacaster is based on a publish/subscribe model. Publishers, or data providers, capture updates on their local databases and propagate those updates to subscribers, or data receivers, who have registered interest in those updates. A subscriber registers its interest in receiving certain data updates by subscribing to a given publisher for a specific, named, set of tables. Various named sets of tables available are defined when metadata is imported to a system.

The following represent the two usage scenarios for the Datacaster in the BPM system:

-
- Step 1** When a Session Controller BPM is brought online to begin handling sessions, it registers with the Policy Repository BPM. The following events occur as part of this registration:
- a. The Policy Repository BPM records the presence of the new Session Controller BPM in its local BPM Registry

- b. The Session Controller BPM uses the *DataCasterAgent* subscribes to the *secureAccess* replication set. The *secureAccess* set contains tables that define the basic aspects of the application, e.g., services, device rules. This set of tables contains data that is by definition the same across all Session Controller BPMs. No subscriber information is part of the *secureAccess* set. A number of tables exist in this set, but the volume of data is not great, and the expected update activity rates are low.

Step 2 When a Director BPM registers with the Policy Repository BPM, it subscribes to the *secureAccessDirector* set, which contains tables pertaining to the mapping of subscribers to Session Controllers, e.g. the *servicedirectoreentry* table. There are only 4 tables in this set, but the volume of data is proportional to the number of subscribers in the system.

When an Session Controller BPM/Director BPM subscribes to a Policy Repository BPM, two events occur:

Step 1 The subscription is recorded in Datacaster-specific tables (in the replication database schema) on both the subscriber (Session Controller BPM/Director BPM) and the publisher (Policy Repository BPM). This information is used by the Policy Repository BPM to identify the data update that subscribers receive as they occur.

Step 2 An initial data copy is made for all tables in the subscribed set. For each table in, for example, the *secureAccess* set the following events occur:

- a. The data in that table on the subscriber is deleted.
- b. The data present in the publisher table is copied to the subscriber table.

The deletion of existing subscriber data and the data copy are configurable via flags passed to the *DataCasterAgent*, but its usage defaults to the scenario described above. The application call to the *DataCasterAgent* *subscribe* method is synchronous, and the *subscribe* operation does not return until the initial data copy is complete.

For a Director BPM subscription, the initial data copy can take time, due to the large volume of data; e.g. one *servicedirectoreentry* record per subscriber. For a large number of subscribers, a substantial amount of data is transferred before the subscription operation is complete.

The Datacaster mechanism does not relate to propagating subscriber-specific data from the Policy Repository BPM to the Session Controller BPM. The application handles this process. When a subscriber logs on for the first time, the Session Controller BPM determines if it has data for that subscriber locally. If it does not, it queries the Policy Repository BPM for the data. The data is propagated from the Policy Repository BPM to the Session Controller BPM, and the Policy Repository BPM notes the relationship between the subscriber and the Session Controller BPM for the benefit of the Director BPMs. When this relationship is established, a record in the *servicedirectoreentry* table is created, and the Policy Repository BPM Datacaster propagates this update to all subscribed Director BPMs.

During normal operations, as updates happen in the publisher database, the Datacaster running on the publisher detects those updates and determines what subscribers, if any, require the new data. For each subscriber for the current data update, the publisher Datacaster connects to the subscriber Datacaster and pushes the data update to it. If a given subscriber Datacaster is unreachable, the publisher retries (a configurable number of times; default: (3)). If the subscriber remains unreachable, that subscriber host is marked as offline. The update transaction destined for that subscriber is retained in the publisher transaction queue (*publishedtransaction* table). The publisher has a thread devoted to checking offline subscribers to detect when they come back online. For example, on the event of a Session Controller BPM failover, during which that Session Controller BPM is temporarily unreachable, the Datacaster

queues updates for that Session Controller BPM subscriber. When the failover operation is complete and the *HostChecker* thread determines that the subscriber has become available, the queued updates are pushed to the subscriber, and normal Datacaster operations for that subscriber will be resumed.

A node can subscribe to another node by issuing a CLI subscribe command locally, or by calling the Datacaster agent running on the local engine. A node subscribes to a named set of tables, defined at the metadata level. When you import the *secureAccessDatabaseGUISupport.tzz* file, you receive the set definition, *secureAccess*, for the tables to be datacasted in phase one. When a node subscribes, it can, optionally, receive an initial copy of all table data in the subscribed set. Once the subscription is complete, the subscribed node receives all updates that occur on its subscribed tables on the publisher.

A service-based subscription is made using a call to the Datacaster agent *subscribe* method. Table 1-1 describes the parameters for this method. If *copyInitialData* is true, the call to *subscribe()* returns only after the full data set is copied to the subscriber.

Table 1-1 **Subscribe Method Parameters and Descriptions**

Parameter	Description
ipAddress	The address of the publisher to subscriber to.
port	The RMI port on the publisher for establishing the RMI connection with the remote datacaster.
clearLocalData	The boolean indicating delete or do not delete local data for subscribed tables.
copyInitialData	The boolean indicating initially copy all existing data from the publisher for the subscribed tables.

During system startup, a registration process occurs in which an SP Session Controller BPM and an SP Director BPM register with the SP Policy Repository BPM. This happens using the *registerPCS* service, which is deployed on both the SP Session Controller BPM and the SP Director BPM. Upon successful registration, the BPM Registry on the SP Policy Repository has a specific configuration entry for both the SP Session Controller BPM and the SP Director BPM. This invokes the Datacaster agent services that move the *secureAccess* tables and the *secureAccessRegistration* tables to the SP Session Controller BPM from the SP Policy Repository. The SP Session Controller BPM moves the *secureAccessUsage* tables to the SP Policy Repository. Upon registration by the SP Director BPM, the SP Policy Repository BPM moves the *secureAccessDirector* table (except for the accessline and subscriber tables) and the *secureAccessRegistration* table to the SP Director.

After a subscriber login (start accounting), the accessline and subscriber tables on the SP Policy Repository BPM, are updated, with the BPM registration ID and are cached to the SP Session Controller BPM. The *serviceDirectorEntry* table on the SP Policy Repository BPM is updated and moved to the SP Director BPM. Once a subscription to the datacaster is made, a subsequent addition, modification, or deletion to a table in the subscribed replication set is cached.

The datacaster process assumes that replication is installed on the publisher and that the *secureAccessDatabaseGUISupport.tzz* is installed on both the publisher and the subscriber. It does not require a special BPDS install.

Initial Data Load

The Datacaster process does not propagate subscriber information. The application propagates subscriber information when a subscriber first logs on.

Return to Initial State

Some conditions exist during which the Datacaster returns to its initial state. Since Datacaster is fully integrated with the platform failover mechanisms, in the Live Upgrade case, Datacaster continues to function as usual. Manual backup and restore procedures include restoring Datacaster subscription tables, so subscription relationships can be maintained through an upgrade. The Datacaster process does not own any application data: it maintains its own internal tables, and moves application data updates from one place to another (publisher to subscriber).

Failure Recovery

During the recovery after a failed Session Controller BPM, a new Session Controller BPM with the base level install replaces the failed Session Controller BPM. This necessitates a new Datacaster subscription because there is no way of knowing what changes have happened in the `secureAccess` or `secureAccessDirector` sets since the Session Controller BPM/Director BPM cluster failed. Any backups done from the Session Controller BPM/Director BPM prior to the failure are probably out of sync. Backup/restore of the Datacaster itself is not relevant, because the Datacaster is a process, not a data repository, with the exception of the tables it uses internally to track subscriptions and transactions.

The system configures the IP addresses and port numbers for this Session Controller BPM. The application schema is installed on the Session Controller BPM by using the BPDS to import the Session Controller BPM and database TZZ package. The agent and service information is recovered using the BPDS reset option. The Director BPM resubscribes to the Policy Repository BPM datacaster and receives the routing table.

During the recovery, the standby system datacaster establishes connections to its subscribers so that the loss of a single node does not affect service. Active transactions from the failed datacaster are rolled back. The loss of an entire cluster *does* prevent new subscribers and uncached subscribers from being subject to policy control. The datacaster retains transactions for the failed Session Controller BPM until it is online.

Quotas

A quota limits volume or service usage and defines a *breach* service to be activated if the subscriber exceeds his quota limits. A quota can specify a network profile for use by that breach service, overriding other profile that the service uses. A quota can receive one or more supplements.

Supplements

A subscriber can purchase a quota supplement, effectively increasing his quota limit. He can continue to purchase quota supplements as long as the resulting effective quota limit is less than the quota supplement limit of the service profile (if the service profile defines a quota supplement limit).

You can set usage thresholds for a quota. When you set a quota threshold, an event notifies an external application when the subscriber reaches the threshold. Usage of bandwidth volume, time, and service invocations is collected and stored in usage records for each subscriber.

Service Bundles and Packages

A service bundle holds all information relating to subscriber services. The Cisco bundle defines the subscriber service package. The Cisco bundle holds all the services a subscriber can use.

The service provider handles subscriber information, including his routing index, line information, his identity at login and his package. When the subscriber logs onto the network, the provisioner knows the subscriber service group and the service bundle that contains his basic service package. When the subscriber is provisioned, the provider provisions his line ID information. The subscriber receives his identifier and his service package, which matches a service bundle on the SP Policy Repository SP Session Controller BPM. This gives him his service options, for example, turbo or the ability to change his service level a specific number of times during a month.

Service bundles also track subscriber quotas. For different data types, and, based on his package, the subscriber may receive a quota for his email or FTP. If he reaches a quota limit, he receives notification to that effect and cannot use the service without purchasing additional quota.

Each service event in a bundle can support a quota that limits service consumption, usage duration, and iterations during a time period. Quota information at the bundle level indicates overall maximum volume limit for a subscriber using the bundle. The quota information at the service event level indicates the level of support a particular service has within a bundle, for example, traffic volume and the time and number of times he can use the service before a quota breach occurs. A service bundle can contain combinations of the following features:

- Initial Login Bandwidth
- Allowed Subscriber Services
- Boost Package Support
- Maximum Usage Values
- Action Taken Upon Volume Quota Breach
- Quota Triggers
- Supplemental Volume Limit
- Quota Resets
- Statistics Tracking

Initial Login Bandwidth

Network profile supports the configuration of the network device to its initial settings upon subscriber login.

Allowed Subscriber Services

Since a service bundle is based on a service profile, the service setup is implemented with service events, as in the service profile.

Boost Package Support

Boost package support allows service events to have separate quota information, if the service profile was set up as a bundle. The separate quota allows tracking of a breach for a specific service. The service event quota contains maximum values for the volume, allowed usage time, and the number of times it can run during the quota time period.

Maximum Usage Values

Maximum usage values use the following allowed volume, duration, and iteration criteria:

- Volume - Total transfer allowed per quota period for a subscriber or service.
- Time - Allowed run time over the quota period.

- Number of Times - Number of times a service can run during a month.

The service bundle supports maximum usage values at the bundle level and the service event level. Only the bundle level has a configuration for maximum volume.

Action Taken Upon Volume Quota Breach

The service bundle supports the ability to program a service that can run when a breach is detected. The service bundle also gives the administrator the ability to provide a network profile that can override the service network profile. The breach service must be a service that contains an event trigger. If a subscriber logs in or out after he breaches his quota, the breach network profile programs the network device with its initial (usually downgraded) bandwidth configuration. To exit a breach, he must apply a supplement or wait for the quota reset.

Quota Triggers

Quota triggers are based on allowed supplemental volume, time, and number of uses. The service bundle gives the administrator the ability to set up supplemental quota (volume for the bundle level and volume, time and number of times for the service event level). A supplement is created separately from the service bundle/service events and *attached* to the bundle/service event.

Supplemental Volume Limits

The service bundle limits the volume supplements a subscriber can use during the quota period. Although a subscriber pays for the supplemental volume, he cannot *infinitely* continue to add volume.

Quota Resets

Quotas can be reset monthly or can be based on a set quota period. Each subscriber has an associated usage table. A new scheduled service is created that can reset all subscriber usage information.

Statistics Tracking

Each subscriber has an associated usage statistics table. Usage statistics are *not* reset when subscriber usage is reset.

Usage Statistics

Usage statistics record traffic amounts by type. Usage statistics tables allow quota and threshold tracking for each subscriber. While most systems are split, using both network and service configurations, a BPM configuration object defines the use of a specific BPM and the BPM devices that provide the network or service functionality for it. Timeouts define the time duration allotted for completion of inter-BPM, intra-BPM and session state activities.

SP Session Controller BPM

In the retail network, the SP Session Controller BPM communicates directly with network resources to provide dynamic control of subscriber sessions and perform resource control functions. The SP Session Controller BPM communicates with other BPM devices using the TRPC protocol. Each SP Session Controller BPM can support up to 120,000 active sessions.

The SP Session Controller BPM knows the bundle names that match the packages available to the subscriber. The bundles contain the differences in the treatment types for the subscriber data and the services he can access. When the subscriber logs onto the network, the SP Policy Repository, which

stores all configurations, receives a request from the SP Session Controller BPM or SP Director BPM. This forces the subscriber onto that Session Controller BPM. The subscriber configuration is cached to the BPM when the system knows that the subscriber is present.

In a simple environment (total subscriber population under 100,000) all functions can be contained in a single SP Session Controller BPM. In large deployments, centralization of subscriber information allows subscriber information to be independent of physical binding to any specific Session Controller BPM. A Director BPM has a route table that correlates live subscriber session to specific Session Controller BPMs.

Login

When a subscriber logs in, he logs into a service bundle. When he logs into the BPM via a RADIUS accounting message, several checks ensure that he can use this Session Controller BPM. When the checks are complete and the results are *valid*, the BPM determines the subscriber service profile. When the service profile is retrieved, a check determines if it is a service bundle. If it is not a service bundle, the system considers this a non-DPI login and the subscriber logs in accordingly.

If this is a service bundle, the BPM device retrieves additional information to configure the DPI device. If this is the first subscriber login, his usage table is populated. The information is read from the quota configuration of the service bundle and any service events with quota information. If the subscriber has logged in before, a check of the service bundle quota configuration determines if it was modified since his last login and the usage table is recalculated.

The BPM determines if the subscriber is in breach. If the subscriber is in breach, the BPM retrieves the breach DPI profile and, if applicable, the breach network profile of the service bundle quota configuration. The BPM records this DPI profile to be set for the subscriber when the BPM logs into the DPI device. If a network profile is retrieved, it is recorded to set up the BRAS device with this profile, later in the login path. If a breach DPI profile is not defined, the default profile, defined at the service bundle level, becomes the profile. If the subscriber is not in breach, the BPM uses the DPI profile and/or the network profile configured at the service bundle level.

After the remaining login path is completed and the subscriber finalized, the BPM logs into the device with the subscriber address, routing index, DPI profile, and the accounting information that the device requires to track usage. When this occurs, the login of the DPI path is complete.

Profile

When the subscriber logs in, the BRAS address obtained from the RADIUS NAS-IP address determines a temporary network profile that limits the subscriber to bare bone access. The temporary network profile is in the access line entry. One access line entry exists per BRAS, where the only elements looked at in the entry will be the BRAS IP address and the network profile that is to be used as the temporary profile for that BRAS address.

The BRAS is set with the initial bare bones profile, which is assigned until the service provider BPM indicates the actual default profile this subscriber receives as part of the package he or she purchased. Every subscriber that logs into a specific BRAS receives the bare bones profile.

The retail side determines the login profile, access group, and network profile. The service session begins and the appropriate subscriber service profile information and appropriate bundles and network profile treatment are determined.

The following subscriber information and default network profile associated with the group exists on the retail side:

- Username
- Domain

- Group
- Service profile/bundle
- Default network profile associated with the group

Accounting Usage Updates

The DPI device sends periodic usage updates to the BPM for any subscribers it is tracking. The DPI device notifies the BPM that a file is ready. The BPM retrieves the files and, for records, retrieves the following information for each subscriber and traffic type:

- subscriber routing index
- traffic type
- bytes sent
- bytes received
- profile name
- subscriber address

Once this update is received and the subscriber exists in the database, the usage statistics table records the traffic type and the *sent* and *received* bytes. This table shows all usage per traffic type, regardless of which DPI profile the subscriber uses. The information screen indicates the traffic type and amount that the subscriber uses on a regular basis. This information can help determine usage trends.

The BPM determines if the subscriber usage table indicates a breach. If the subscriber is in breach, the BPM does not record the usage information. If the subscriber is not in breach, the BPM calculates the updated usage information and checks again for a breach condition. If the updated usage is not in breach, the BPM reads additional information from the file.

If the subscriber is in breach, the BPM retrieves the breach profile information from the service bundle quota configuration. If information exists, the BPM logs into the DPI device with the new DPI profile (if a DPI profile exists) and sends a new network profile to the BRAS device on the networks side, (if a network profile exists). The BPM then returns to read the next record from the accounting file.

Breach Notifications

Since the accounting update files take time to upload, the DPI device generates a near real-time notification of a subscriber breach condition. When the DPI device detects a subscriber breach, the BPM receives a message indicating the breach condition. The message includes the subscriber routing index, DPI profile name, current usage, direction of current usage, and subscriber address.

Unlike the accounting usage update, the breach notification does not indicate bytes sent and received in the message. It sends the total usage from initial login to the time of the breach and sends this as a total aggregate number to the BPM. It combines the received and sent before sending the BPM the message.

This message runs through much of the same code path as the accounting usage update message. Since no traffic type information is supplied in this notification, the usage statistics table cannot be updated; so this step is not taken.

When the usage table is updated, the received or sent fields cannot be set with new information since the BPM received only an aggregated number. Also, since this aggregated number is not a delta of usage from the last update, it becomes the total usage field in the usage table. This maintains the accuracy and consistency of information between the BPM and the DPI devices.

The BPM determines from its table if the subscriber is in breach. In breach conditions, it follows the same logic as the accounting usage update.

Volume Supplement

When the supplement volume service runs, a number, in bytes, is passed as the value for increasing the subscriber quota. The subscriber information is retrieved, and the supplement volume passed is verified against the supplement table in the service bundle quota configuration. If the value passed is valid, a check occurs to ensure that this supplement (in addition to other supplements) does not exceed the quota supplement limit defined in the service bundle. If this is valid, the subscriber usage record is updated with this new information.

This supplement volume service can be run, regardless of the subscriber log-in status. If the subscriber has an active session, a check determines if he is in breach before the supplement. If the subscriber is not in breach after the supplement application, any breach profiles applied to the subscriber are removed and the previous profiles are reinstalled.

In a breach condition, the DPI and network profiles of the service bundle are used for subscriber login to the DPI device and a BRAS, if one is configured.

Reset Subscriber Quotas/Monthly Rollover

Rolling over or resetting subscriber quotas occurs through the *resetQuotasByDomain* service. This service can be scheduled to recur, for example, on the last day of every month at 12 a.m. When this service is invoked, all subscribers in the domain are retrieved. Usage for each is zeroed out and the quota limit is reset to that retrieved from the appropriate service bundle. Unused supplement volume is rolled over into the new usage table as part of the new total quota. If a subscriber is logs in during the usage reset process, a check determines if he is in breach. If he is in breach, the logic described in the supplement section occurs.

Administrator Services

The following services are administrator services:

- *02_adminDPIDevice*
- *resetQuotasForDomain*

The *02_adminDPIDevice* service is described above in the Keepalive and Synchronization Mechanism section. The *resetQuotasForDomain* service is described above in the Reset Subscriber Quotas/Monthly Rollover section.

Wholesale Network

The wholesale network components, on the left in [Figure 1-2](#), manage requests for application in the wholesale network. They also and control key contention points within the wholesale network. The wholesale network offers a numbers of request types to the retail application layer.

The following are the wholesale components:

- NP Director BPM
- NP Policy Repository BPM
- NP Session Controller BPM

NP Director BPM

In the wholesale network, the NP Director BPM has a mapping of the NP Session Controller BPM (talking to BRASs) and SP Session Controller (talking to DPI devices). This allows the wholesale network to pass the appropriate information to the retail network, without knowledge of a subscriber (or IP session) or the services that the SP offers.

NP Policy Repository BPM

In the wholesale network, the NP Policy Repository BPM handles database requests from NP Session Controller BPMs, SP Session Controller BPMs, and Director BPMs. As the central database for the Session Controller BPMs, the NP Policy Repository BPM component stores personal subscriber and configuration data. It holds the data required for normal operation, such as policies and rules, service package definitions, identifier mappings, usage statistics, and logs. Each Session Controller BPM has its own disk space and caches the data required for its operation to minimize calls to the Policy Repository.

The retail Policy Repository supports the following transaction volumes on each interface:

Queries from SP Session Controller BPMs: 1000 per second

When a subscriber is provisioned, the Access Line ID and an identifier are created on the Policy Repository BPM. The identifier contains the core of the subscriber information. The unique subscriber identifier is used on the network, instead of a username, domain, and password. Only the two endpoints of the network know the subscriber identity. In this way, his identity never traverses the network.

NP Session Controller BPM

In the wholesale network, the NP Session Controller BPM communicates directly with network resources such as 10K BRAS devices and retail resources to provide dynamic control of subscriber sessions and perform resource control functions. The NP Session Controller BPM communicates with other BPM devices using the TRPC protocol. Each NP Session Controller BPM can support up to 120,000 active sessions.

The NP Session Controller BPM understands the interface and 10K device that the subscriber arrives from. It determines the set of policies for each subscriber session. It knows the default subscriber profile on that 10K device and the services available to him.

.10K BRAS Device

The BSM application integrates with the 10K BRAS device. The integration with the 10K server provides service integration capabilities to dynamically control sessions on the 10K server using RADIUS and CLI commands. This capability allows the service provider to instantiate subscriber service request changes.

RADIUS Server

A RADIUS server provides authentication. RADIUS traffic moves directly from the 10K BRAS device to the Session Controller BPM. RADIUS accounting start, stop, and interim are supported. The start, stop, and interim messages, which require acknowledgements, identify subscriber sessions coming onto, and off of, the network.

The *radiusStartAcct* flow intercepts a RADIUS message to the subscriber and sets up required network devices and policies. It also sets required information in the database. The flow intercepts a RADIUS message and calls the *startAccountingPF* policy function. A fork determines if the response is valid. If not, a response is sent to the subscriber. Otherwise, a job is scheduled and the *enablementPF* policy function is called. A response is sent to the subscriber.

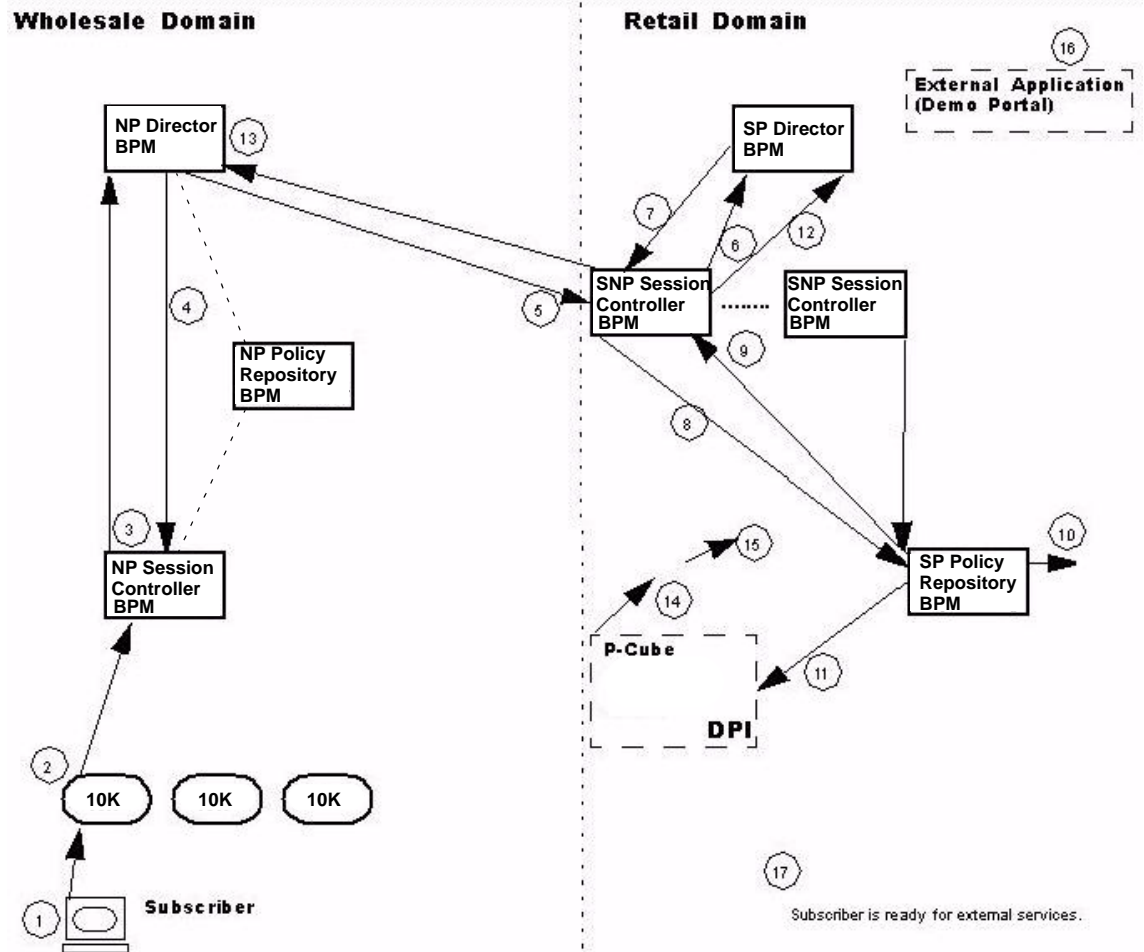
The *radiusStopAccountService* flow intercepts a logout from a RADIUS request and reacts accordingly. Database information is reset or removed when a subscriber logs out. This flow begins with a RADIUS stop accounting message. Once the BPM platform intercepts the message, the flow of this service enters the *stopAccountingPF* policy function. Depending on the output of this policy function, this flow has two possible outcomes:

- Send a response, if this is a typical logout and the BPM information is to be cleared in the process.
- Logs the subscriber out via an *Accept Only Logout* message.

An error occurs when the output from the stopAccountingPF policy function is incorrect.

Figure 1-3 depicts the process.

Figure 1-3 Step-by-Step Details.



Step Details. (Rough Cut)

-
- Step 1** The Subscriber begins his login via 10K BRAS device. 10K BRAS device intercepts login and sends to NP Session Controller BPM. (RADIUS; Here, applies network policy?)
- Step 2** NP Session Controller BPM sends a *serviceSessionBegin* message to the NP Director BPM.
- Step 3** NP Director BPM sends an ack/nack response to NP Session Controller BPM.
- Step 4** NP Director BPM - communicates with the SP Session Controller BPM. The NP Director BPM receives the forwarding table and sends it to SP Session Controller BPM. The SP Session Controller BPM queries SP Director BPM to determine if this subscriber is known by the SP Director BPM, which has the IP addresses for the subscribers. If the subscriber is known, the SP Director BPM informs the NP Director BPM. If the subscriber is not known, SP Director BPM queries SP Policy Repository. The subscriber IP address, userid (username@domain) and session id are learned during this exchange.
- Step 5** When the SP Session Controller BPM receives a *serviceSessionBegin* message from the NP Director BPM or NP Session Controller BPM, it determines if the subscriber is already registered with the SP Director BPM.
- Step 6** The SP Director BPM responds.
- Step 7** If it is not registered with the SP Director BPM, the SP Session Controller BPM determines if an external id exists in the subscriber policy database entry Policy. The SP Policy Repository holds all configuration for all network profiles. The SP Policy Repository checks for subscriber info. The SP Policy Repository disseminates all profiles.
- Step 8** If this id does not exist, the SP Controller BPM allocates an external device identifier and communicates the appropriate information to the SP Director via the SP Policy Repository BPM.
- Step 9** The SP Policy Repository BPM communicates that information to the other SP Policy Repository BPMS. Datacaster and updates all SP Policy Repository BPMS of subscriber information. The SP Policy Repository BPM also sends out information to all SP Policy Repositories. (The GUI originally populates the SP Policy Repository BPM and NP Policy Repository BPM.) Datacaster then updates all SP Policy Repository BPMS of subscriber information.
- Step 10** SP Policy Repository finds subscriber information and informs DPI. (Profile pushed to DPI?) The DPI device, although owned by the retail domain, sits with 10K BRAS device information. to NP Director BPM, installs profile.
- Step 11** Installs profiles to NP Director BPM. SP Director BPM learns where to direct information for the subscriber.
- Step 12** SP Session Controller BPM applies service policies. It stores baseline configuration and profile information in the temporary temp session. Tracking begins.
- Step 13** If OK, the subscriber receives basic network profile and service profile. The login session is active. The subscriber is ready for services.

Secure Access Model

The Secure Access model and the BSM application support the following BPM wholesale (network provider) devices:

- NP Director BPM
- NP Session Controller BPM
- NP Policy Repository BPM

The BSM application supports the following BPM retail (service provider) devices:

- SP Director BPM
- SP Session Controller BPM
- SP Policy Repository BPM

The network provider (NP) applications represent the BPMs that support the wholesale capability. The service provider (SP) applications represent the BPMs that support the retail capability.

The NP Policy Repository BPM contains the set of services and policy rules that support the Cisco connection to the 10K BRAS devices. The NP Policy Repository BPM stores all commonly configured data for the NP BPMs. The NP Policy Repository BPM performs the initial configuration of all NP Session Controller BPM that it connects to and for forwarding any changes to all NP BPMs.

The NP Director BPM determines the NP Session Controller BPM and SP Session Controller BPM a subscriber is associated with. The SP Session Controller BPM and NP Session Controller BPMs communicate directly with the NP Director BPM using, at a minimum, the NAS IP address. The NAS IP address determines which Network or Service BPM the subscriber is associated with. The NP Director BPM is populated by the NP Policy Repository BPM.

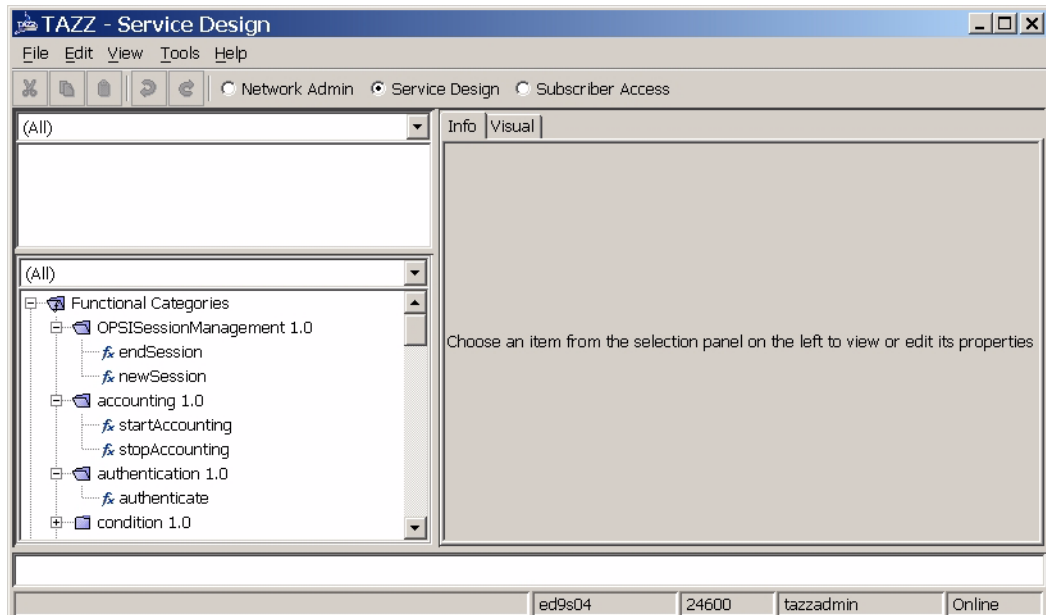
The SP Policy Repository BPM contains the services and policy rules to support the Cisco connection to external retail applications, DPI devices, and the NP Director BPM. The SP Policy Repository BPM stores all commonly configured data for the SP BPMs. The SP Policy Repository BPM initially configures all the SP BPMs that it connects to and forwards any data changes to those SP BPMs.

When an external application must effect a change to a subscriber or obtain subscriber information, the SP Director BPM determines the SP Session Controller BPM the subscriber is associated with. External applications communicate directly to the SP Director BPMs through a SOAP interface and have a registration mechanism for determining a unique identifier for each subscriber the external application must access. The SP Director BPM is populated by the SP Policy Repository BPM and the entries are created in the SP Policy Repository BPM at initial subscriber login. After the initial subscriber login, a subscriber director entry is updated only if the subscriber is moved to a 10K BRAS device associated with a different SP BPM.

Broadband Policy Design Studio

The service provider administrator uses the BPM Broadband Policy Design Studio (BPDS) to manage BPM components. The BPDS connects directly to provide secure, real-time control over all aspects of retail and wholesale service operations. The administrator can also use the BPDS to develop additional BPM components to automate manual processes.

Figure 1-4 Typical BPDS Screen.



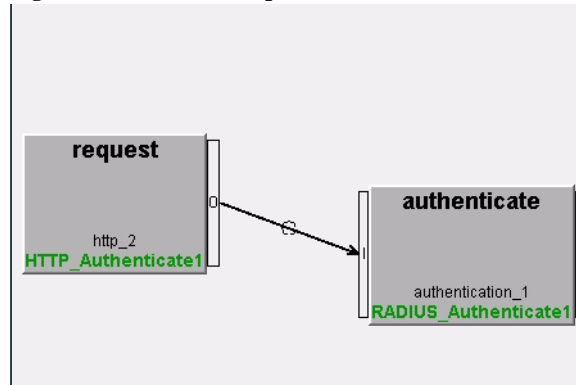
With the BPDS, the administrator can develop additional BPM components, which comprise flows, services, rules, policy functions, agents, device types, device rules, and device profiles. This section discusses the following BPM components:

- [Flows](#)
- [Services](#)
- [Rules](#)
- [Policy Functions](#)
- [Agents](#)
- [Device Types](#)
- [Device Rules](#)
- [Device Profiles](#)

Flow

In the BPM system, a *flow* represents movement of data or control and is a basic element of a *service*. The service developer uses those flows to define services. [Figure 1-5](#) presents a typical flow.

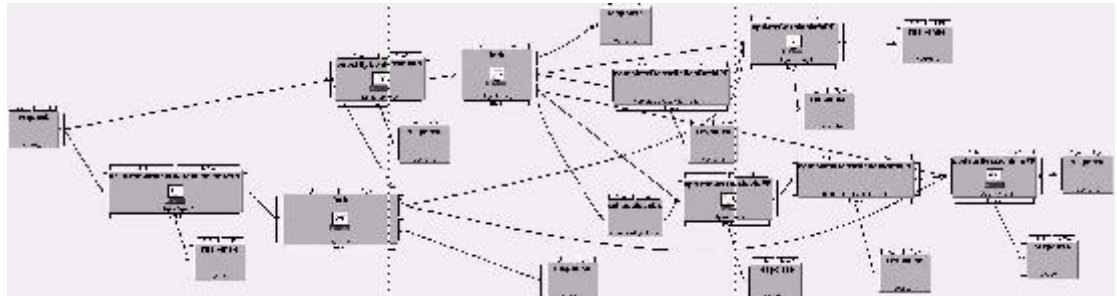
Figure 1-5 A Simple Flow.



Service

A simple *service* might contain one flow. However, a service can contain multiple flows. Figure 1-6 presents a service with multiple flows.

Figure 1-6 A Sample Service.



Once a service is created, the developer can use it as part of larger, more complex, services. The developer uses flows and services to develop *applications*. The service or application maps business models or operational procedures directly into IP services, executable by a service provider. For example, a service could execute a video on demand application or schedule an automatic system backup.

Rule

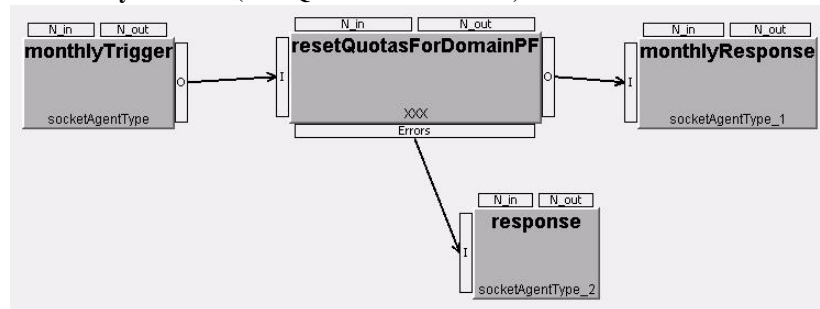
A *rule* or set of rules contains criteria that apply to the methods of a business system. Rules determine how flows or services within the BPM system are used by, or for, a given system subscriber. One rule can invoke another rule, and rules can become *policy functions*.

Policy Function

A *policy function* is a flow containing a rule or set of rules that can perform a specific action. For example, a policy function for subscriber access could direct how the system identifies a subscriber, using his userid, access type, and login location.

A policy function performs an operation, based on input, and returns the results of its action as output. Rules and policy functions, developed and used in flows, can represent policies that further control access to network resources. Policy functions used in flows can invoke other functions or trigger other flows. [Figure 1-7](#) presents a policy function, *resetQuotasForDomainPF*, used in a service.

Figure 1-7 A Policy Function (*resetQuotasForDomainPF*) Used in a Service.



Agent

An *agent* interacts with a particular device type. For example, a RADIUS agent package contains software that allows the developer to create agents that interact with specific RADIUS devices. A service uses specific agents and the functions they provide. An agent interacts with a device, and the agent functions perform the service operations, based on incoming parameters, and return the results to the agent. See [Chapter 6](#) for more agent information.

Device Types

A *device type* is a classification of an actual device, based on device attributes, such as vendor, model, and hardware and software versions. Device types created with the BPDS identify the hardware devices that the BPM platform communicates with. Several preconfigured device types exist for the BPM. The following list contains devices preconfigured for the BPM platform:

- Cisco 10000 (2/12.2)
- Cisco 2610 (14.3/12.4)
- Cisco 7206 (VRS/12.2)
- P-Cube SE1000 (3.2/4.0)

Device Rule

A *device rule* defines actions performed on a device. It represents a generic task for application on a device that supports the device rule functionality for a specific action. Device rules can retrieve information from connected devices. Preconfigured device rules exist for configuring a new BPM platform. Most device rules assume that appropriate policies are defined on the network hardware. See [Chapter 3](#) for a further discussion of device rules.

Device Profile

A *device profile* executes a series of device rules when specific conditions are met, for example, when a subscriber logs on, logs off, or changes his bandwidth. Predefined device profiles exist for use in new BPM platform deployments. Each predefined profile has install and uninstall device rules. See Chapter 3 for a further discussion of device profiles.



Configuration

Overview

This chapter discusses the configuration of the Broadband Policy Manager (BPM) devices to implement specific functionality for the wholesale and retail sides of the network. The configuration defines the BPM device role and determines the locations of its resources. Some resources are on the BPM device. Other resources, such as remote data stores and network, service, and policy repositories are on other BPM devices. The appropriately configured devices interact with each other and with other network components to provide the Broadband IP Services Module (BSM) solution.

The following are the possible BPM device configurations:

- Director BPM
- Policy Repository BPM
- Session Controller BPM

The type entry in a BPM configuration must agree with the application (services and agents) loaded on the target BPM device. The type signals the Broadband Policy Design Studio (BPDS) to provide the appropriate view into the database tables for this BPM device.

The network BPM entry and the service BPM entry on the BPM configuration have the host and port of the BPM device that handles the specified request type for this BPM device. If this BPM device has a network configuration, the host and port of the network BPM device in the BPM configuration have the host name (or IP address) and port of this BPM device. The service BPM host and port have the host name (or IP address) and port of another BPM device with the service configuration that handles service requests for this BPM device. If this BPM device has a service configuration, the host and port of the service BPM device in the BPM configuration have the host name (or IP address) and port of this BPM device.

The network BPM host and port have the host name (or IP address) and port of another BPM device with a network configuration that handles network requests for this BPM device.

The URL field for both network and service BPM devices represents the name (servicename/flowname) of the service on a specific BPM device that processes requests from other BPM devices. The default URL for a service BPM device is *processNtwkEvent/processNtwkEvent*. The default URL for a network BPM device is *processServiceEvent/processServiceEvent*. When the BPM devices in a given system use these defaults, the URL fields can be empty. However, if you rename these services, you must enter the name of the new URL here. All arguments to these services must remain the same. Only the name of the service or flow may change, if these services are to function correctly.

When a BPM device with a director configuration handles requests for more than one BPM device, the Service BPM URL is *processServiceEventDirector/processServiceEventDirector* and the Network BPM URL is *processNtwkEventDirector/processNtwkEventDirector*.

A Policy Repository BPM device acts as a central data store from which other BPM devices can cache a subset of the repository data. A repository does not act as a network or service BPM device, nor does a repository use another repository BPM device. These are not configurable once the repository check box is selected.

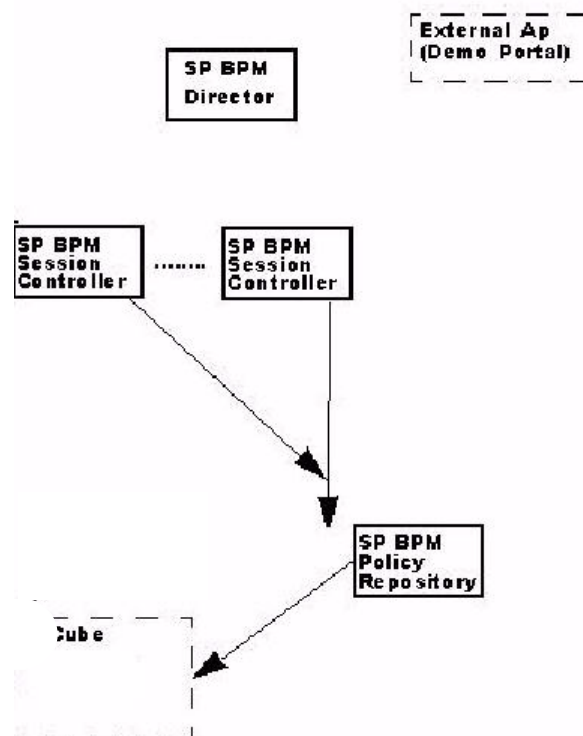
Retail Domain Devices

This section discusses configuration of the devices on the retail side of the BPM network:

- SP Director BPM
- SP Policy Repository BPM
- SP Session Controller BPM

Figure 1-1 presents the retail domain network components.

Figure 1-1 Retail Domain
Retail Domain



Director BPM

A Director BPM routes requests to other BPM devices. When the director receives a request, it locates the BPM device that contains the information for a given subscriber or access line and forwards the request to that BPM device.

You can add the BPM configuration by using the **New** menu option or by importing a policy data file that contains the appropriate BPM configuration. However, do not use the **New** menu option on a system that you plan to import policy data onto. If the imported data contains a BPM configuration, the BPM device detects and adjusts to it. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until you add a BPM configuration using the **New** menu option.

The Director BPM configuration contains rules and policy functions that determine how the data objects in the configuration interact. This configuration can schedule and can grant access to service for administrators and subscribers.

Procedure: Configure the SP Director BPM

To configure the SP Director BPM for the retail network, perform the procedure below.

-
- Step 1** Log on as the administrator.
 - Step 2** Load the backend software into the `/home/tazzadmin/software/tazzxxxx/` directory, making sure the transfer is done in binary mode.
where
tazzxxxx is the directory you designate.
 - Step 3** Install backend in `[/home/tazzadmin/software/tazz1405/tazz]: /tazz`
 - Step 4** Select the Installation type (stand-alone | custom) `[stand-alone]: custom`
 - Step 5** Select the Install Service Engine (yes or no) `[yes]: yes`
 - Step 6** Select the Install Application Database (yes or no) `[yes]: yes`
 - Step 7** Select the Number of database connections `[16]: 48`
 - Step 8** Load the new backend by entering the following commands:

```
cd /home/tazz/software/tazz140x/  
./backend-1xx.sh
```
 - Step 9** Import the `secureAccessDatabaseGUISupport.tzz` file.
 - Step 10** Set the system type as Service Director.
 - Step 11** Change the connection on the BPM device, and use the first Director IP as the application database.
 - Step 12** Click the Subscriber Access radio button, and install the service `Director MetaData` package (usually `directorDatabaseGUISupport.tzz`).
 - Step 13** Select New --> File --> BPM Configuration and enter the following:
 - Step 14** Select `ServiceDirector` under Type.
Local Communications Port is 10100
 - Step 15** Under Policy Repository, enter the following:
Host: 99.99.120.142
Port: 10100
URL: `processSession ControllerRegistration/processSession ControllerRegistration`
 - Step 16** Select Network Admin, right-click the director, and import the service director system package.

Policy Repository BPM

You can add the configuration by using the **New** menu option or by importing the policy data file that contains the appropriate BPM configuration. However, do not use the **New** menu option on a system that you plan to import policy data onto.

If the imported data contains a BPM configuration, the BPM device detects and adjusts to it. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until you add a BPM configuration using the **New** menu option.

This configuration contains rules and policy functions that determine how the data objects in the configuration interact. This configuration can schedule and grant access to services on a specific BPM device for administrators.

Before the metadata is loaded, the options under the *New* option from the *File* menu or from the navigator tree allow the creation of a rule or a policy function. Once the metadata is loaded, this option is expanded for the creation of a BPM configuration.

Procedure: Configure the SP Policy Repository BPM

To configure the SP Policy Repository BPM for the retail network, use the procedure below.

-
- Step 1** Log on as the administrator.
- Download the backend software into the `/home/tazzadmin/software/tazzxxxx/` directory, making sure the transfer is done in binary mode.
- where
- tazzxxxx is the directory you designate.
- Step 2** Install backend in `[/home/tazzadmin/software/tazz1405/tazz]: /tazz`
- Step 3** Load the new backend by entering the following commands:
- ```
cd /home/tazz/software/tazz140x/
./backend-1xx.sh
```
- Step 4** Select the Installation type (stand-alone | custom) `[stand-alone]: custom`
- Step 5** Select the Install Service Engine (yes or no) `[yes]: yes`
- Step 6** Select the Install Application Database (yes or no) `[yes]: yes`
- Step 7** Select the Number of database connections `[16]: 96`
- Step 8** Import the `secureAccessDatabaseGUISupport.tzz` file.
- Step 9** Set the system type as Network.
- Step 10** Click the Subscriber Access radio button, and install the service MetaData file, `combinedDatabaseGUISupport.tzz`.
- Step 11** Select File --> New --> BPM Configuration.
- Step 12** Check the repository checkbox, and enter the local communication port: 10100.
- Step 13** Select Network Admin, highlight the Policy Repository, and import the `policy repository` package.



## Session Controller BPM

You can add the BPM configuration by using the **New** menu option or by importing a policy data file that contains the appropriate BPM configuration. However, do not use the **New** menu option on a system that you plan to import policy data onto.

If the imported data contains a BPM configuration, the BPM device detects and adjusts to it. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until you add a BPM configuration using the **New** menu option.

This configuration contains rules and policy functions that determine how the data objects in the configuration interact. This configuration can schedule and grant access to services on a specific BPM device for subscribers and administrators.

### Procedure: Configure the SP Session Controller BPM

To configure the SP Session Controller BPM for the retail network, use the procedure below.

- 
- Step 1** Log on as the administrator.
- Step 2** Download the backend software into the `/home/tazzadmin/software/tazzxxxx/` directory, making sure the transfer is done in binary mode.
- where  
tazzxxxx is the directory you designate.
- Step 3** Load the new backend by entering the following commands:
- ```
cd /home/tazz/software/tazz140x/  
./backend-1xx.sh
```
- Step 4** Install backend in `[/home/tazzadmin/software/tazz1405/tazz]: /tazz`
- Step 5** Select the Installation type (stand-alone | custom) `[stand-alone]: custom`
- Step 6** Select the Install Service Engine (yes or no) `[yes]: yes`
- Step 7** Select the Install Application Database (yes or no) `[yes]: yes`
- Step 8** Select the Number of database connections `[16]: 48`
- Step 9** Load the new backend by entering the following commands:
- ```
cd /home/tazz/software/tazz140x/
./backend-1xx.sh
```
- Step 10** Take all the defaults in the script, except from the following:
- Step 11** Import the `secureAccessDatabaseGUISupport.tzz` file.
- Step 12** Set the system type as Service.
- Step 13** Click the Subscriber Access radio button, and load metadata into the Session Controller BPM.
- Step 14** Select File --> New --> BPM Configuration.
- Step 15** Configure this as shown below. This procedure below assumes that 99.99.120.142 is the Primary IP of the cluster.
- Remote Data Stores:  
type: service

mode: Non-Cisco Network  
 Local Communication Port: 10100  
 Network BPM Host: localhost  
 Network BPM port: 10110  
 Network BPM URL: leave blank  
 Service BPM Host: localhost  
 Service BPM Port: 10100  
 Service BPM URL: leave blank  
 Event Type: SOAP  
 Under Remote Data Stores, add the following:  
 Name: Session - Host: 99.99.120.142 - Port: 10005

## Procedure: Configure Session Controller Agents

Configure the agents as shown below. If the agent is not on this list, no configuration is required.

- 
- Step 1** Configure the NonTazzCommSocket agent.  
 Under Properties:  
 additionalHostsLists: ANY  
 port: 10110
- Step 2** Configure the ProxyRadius agent.
- Step 3** Under Devices, edit the assigned device (if none, assign AccountServer device):  
 serverPortAccounting: 1813  
 serverSharedSecretAccounting:
- Step 4** Under Devices, edit the assigned device:  
 WSDLURL: http://99.99.120.140/accountingEventMonitor.wsdl  
 (This assumes that 99.99.120.140 is the machine where the wsdl file for a clustered pair is located.)  
 domain: BPM  
 password:  
 retryTimeout: 0  
 user: admin
- Step 5** Configure the SoapAgent agent.  
 Under Properties:  
 port: 10200  
 serviceName: accountingEventMonitor
- Step 6** Configure the event agent.
- Step 7** Under Devices, edit the assigned device (if none, assign Event10000 device):  
 dispatcherHost: localhost  
 dispatcherPort: 10012

**Step 8** Configure the http agent.

Under Properties:

servletAddress: 99.99.120.140

servletPort: 5555

templatePrefix: http://99.99.120.140

urlPrefix: http://99.99.120.140:8080/tazz/servlet/RequestHandler/

(This assumes that 99.99.120.140 is the machine where Apache and Tomcat for a clustered pair is located.)

**Step 9** Configure the scheduler agent.

Under Properties:

socketPort: 10100

Under Devices:

dispatcherHost: localhost

dispatcherPort: 10012

**Step 10** Configure the socket agent.

Under Properties:

additionalHostsList: ANY

port: 10100

**Step 11** Configure the tcl agent.

No configuration.

**Step 12** Edit the Service 30\_soapCommunicationsNonTazz service.

**Step 13** Deploy all agents and services.

**Step 14** Edit the `accountingEventMonitor.wsdl` file

This file is created in `/tazz-install/wsdl` when the services are deployed. Edit this to include the primary IP address of the Session Controller BPM, and then move it to the `/var/apache/htdocs` directory on 99.99.120.140 (or to the location the agent points to). When setting a second Session Controller BPM, do not use the same `accountingEventMonitor.wsdl` file as another Session Controller BPM.

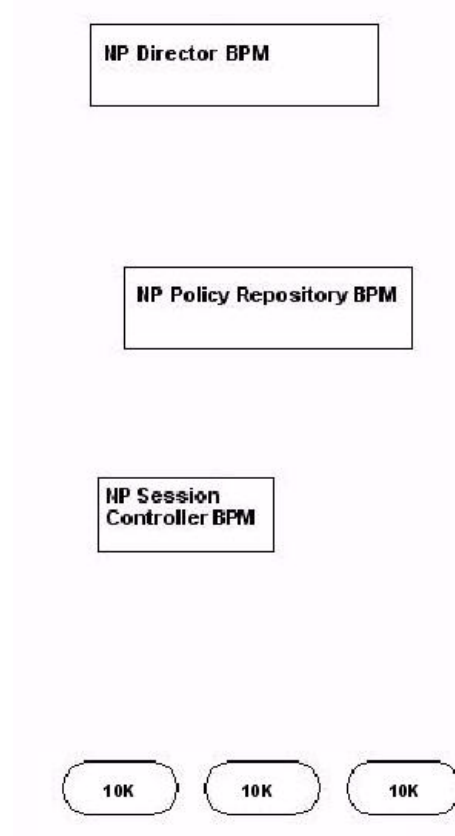
## Wholesale Domain Devices

This section discusses configuration of the following BPM devices on the wholesale side of the BPM network:

- NP Director BPM
- NP Policy Repository BPM
- NP Session Controller BPM

Figure 1-2 presents the wholesale domain network components.

Figure 1-2 Wholesale Domain



## Policy Repository BPM

You can add the Policy Repository BPM configuration by using the **New** menu option or by importing the policy data file that contains the appropriate BPM configuration. However, do not use the **New** menu option on a system that you plan to import policy data onto.

If the imported data contains a BPM configuration, the BPM device detects and adjusts to it. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until you add a BPM configuration using the **New** menu option.

This configuration contains rules and policy functions that determine how the data objects in the configuration interact. This configuration can schedule and grant access to services on a specific BPM device for administrators.

### Procedure - Configure the NP Policy Repository BPM

To configure the NP Policy Repository BPM for the wholesale network, perform the procedure below.

- 
- Step 1** Log on as the administrator.
  - Step 2** Download the backend software into the `/home/tazzadmin/software/tazzxxxx/` directory, making sure the transfer is done in binary mode.

where

tazzxxxx is the directory you designate.

- Step 3** Select the Installation type (stand-alone | custom) [stand-alone]: custom
- Step 4** Select the Install Service Engine (yes or no) [yes]: yes
- Step 5** Select the Install Application Database (yes or no) [yes]: yes
- Step 6** Select the Number of database connections [16]: 96
- Step 7** Import the *secureAccessDatabaseGUISupport.tzz* file.
- Step 8** Set the system type as Network.
- Step 9** Click the Subscriber Access radio button, and install the service MetaData file, *combinedDatabaseGUISupport.tzz*.
- Step 10** Select File --> New --> BPM Configuration.
- Step 11** Check the repository checkbox, and enter the local communication port: 10100.
- Step 12** Select Network Admin, highlight the Policy Repository, and import the *Policy Repository* package.

Before the metadata is loaded, the options under the *New* option from the *File* menu or from the navigator tree allow for the creation of a rule or a policy function. Once the metadata is loaded, this option is expanded for the creation of a BPM configuration.

## Session Controller BPM

You can add the BPM configuration by using the **New** menu option or by importing a policy data file that contains the appropriate BPM configuration. However, do not use the **New** menu option on a system that you plan to import policy data onto. If the imported data contains a BPM configuration, the BPM device detects and adjusts to it. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until you add a BPM configuration using the **New** menu option.

This configuration contains rules and policy functions that determine how the data objects in the configuration interact. This configuration can schedule and grant access to services on a specific BPM device for administrators.

### Procedure: Configure the NP Session Controller BPM

To configure the NP Session Controller BPM for the wholesale network, for the wholesale network, use the procedure below.

- 
- Step 1** Log on as the administrator.
  - Step 2** Download the backend software into the `/home/tazzadmin/software/tazzxxxx/` directory, making sure the transfer is done in binary mode.  
where  
tazzxxxx is the directory you designate.
  - Step 3** Install backend in `[/home/tazzadmin/software/tazz1405/tazz]: /tazz`
  - Step 4** Select the Installation type (stand-alone | custom) [stand-alone]: custom
  - Step 5** Select the Install Service Engine (yes or no) [yes]: yes

- Step 6** Select the Install Application Database (yes or no) [yes]: yes
- Step 7** Select the Number of database connections [16]: 48
- Step 8** Import the *secureAccessDatabaseGUISupport.tzz* file.
- Step 9** Set the system type as Network.
- Step 10** Click the Subscriber Access radio button, and load metadata into the Session Controller BPM.
- Step 11** Select File --> New --> BPM Configuration.
- Step 12** Configure this as follows (the procedure below assumes that 99.99.120.142 is the Primary IP of the cluster):
- Remote Data Stores:
- type: service
- mode: Non-Cisco Network
- Local Communication Port: 10100
- Network BPM Host: localhost
- Network BPM port: 10110
- Network BPM URL: leave blank
- Service BPM Host: localhost
- Service BPM Port: 10100
- Service BPM URL: leave blank
- Event Type: SOAP
- Step 13** Under Remote Data Stores, add the following:
- Name: Session - Host: 99.99.120.142 - Port: 10005
- Name: Policy - Host: 99.99.120.142 - Port: 10005
- Step 14** Under Repository BPM, add the following:
- Host: 99.99.120.142 - Port: 10100
- Under Caching, add the following:
- processServiceCacheRequest/processServiceCacheRequest
- Local Devices
- Step 15** Model/Vendor Allowed Values, Device Type, and Device creation must occur in the Session Controller. The Device Type must be an exact match with the Device Type stored on the Policy Repository.
- Step 16** Click Submit to save settings.
- Step 17** Select *Network Admin*, highlight the first Session Controller, and import the service system .tzz package.



# System Specific Services

---

## Overview

This chapter organizes Broadband IP Services Module (BSM) services as supported by the following systems:

- NP Director BPM
- SP Director BPM
- NP Session Controller BPM
- SP BPM
- 

## SP Director BPM

This SP Director BPM system supports the following services:

- 01\_adminServiceTrigger
- 03\_refreshSCPCConfig
- 04\_subscribeToDatacaster
- 05\_addTimerToScheduler
- alterProfile
- applyProfile
- applyProfileFeature
- getSubscriberExternalIndex
- registerApplication
- registerPCS
- releaseProfile
- sendPCSRegistration
- sendRegisterApplication
- sendUnregisterApplication
- subscriberProcessProfileRequest

- subscriberProcessProfileResponse
- unregisterApplication
- updateSPDirectorEntry

## NP Director BPM

The NP Director BPM system supports the following services:

- 03\_refreshSCPCConfig
- deliverSubscriberNetworkEvent
- deliverSubscriberServiceEvent
- subscriberNetworkToServiceEvent
- subscriberServiceToNetworkEvent

## NP Session Controller BPM

The NP Session Controller BPM system supports the following services:

- 01\_adminServiceTrigger
- 03\_refreshSCPCConfig
- 04\_subscribeToDatacaster
- 05\_addTimerToScheduler
- deviceConfig
- deviceConfigISG
- genericNetworkStateMachine
- genericRadiusStartAccounting
- genericRadiusStopAccounting
- registerPCS
- sendCacheRequest
- sendPCSRegistration
- subscriberNetworkProcessServiceEvent
- subscriberNetworkToServiceRequest

## SP Session Controller BPM

The SP Session Controller BPM system supports the following services:

- 01\_adminServiceTrigger
- 02\_adminDPIDevice
- 03\_refreshSCPCConfig



- 04\_subscribeToDatacaster
- 05\_addTimerToScheduler
- 20\_adminMgmt
- bundleActivationEvent
- genericServiceStateMachine
- pcubeDPISubscriberLogin
- pcubeDPIScriberLogoff
- pcubeDPISubscriberUsage
- quotaBreach
- registerPCS
- registerSubscriberWithDirector
- resetQuotasForDomain
- revertQuotaBreach
- scheduledStateMachineEntry
- sendCacheRequest
- sendPCSRegistration
- serviceProfileEventMgmt
- serviceProfileSessionUpdate
- subscriberProcessProfileEvent
- subscriberServiceProcessNetworkEvent
- subscriberServiceToNetworkRequest

## NP Policy Repository BPM

The Network Policy Repository BPM system supports the following services:

- 03\_refreshSCPCfg
- processnetworkCacheRequest
- notifyAccessLineChange

## SP Policy Repository BPM

The SP Policy Repository BPM system supports the following services:

- 01\_adminServiceTrigger
- 03\_refreshSCPCfg
- 20\_adminMgmt
- generateStatisticFilesPR
- serviceProfileEventMgmt
- serviceProfileSessionUpdate

- spProvisionHttpNotification
- spProvisionMissingAccessline
- spProvisionServiceProfileChange
- spProvisionSIDQuery
- processServiceCacheRequest
- processBPMRegistration



# Deployment

---

## Overview

This chapter discusses Broadband Policy Manager (BPM) nonregulated and regulated deployment, its components, policy management, and supported traffic classes and services. The BPM supports a carrier-class environment and delivers high performance and resiliency. It offers standard interfaces to facilitate integration with the most prevalent systems interfaces. This architecture offers both integration and business model flexibility. The BPM architecture, based upon a deployed Cisco-regulated wholesale/retail carrier architecture, supports both non-regulated and regulated deployments.

This chapter discusses the following BPM application deployment topics:

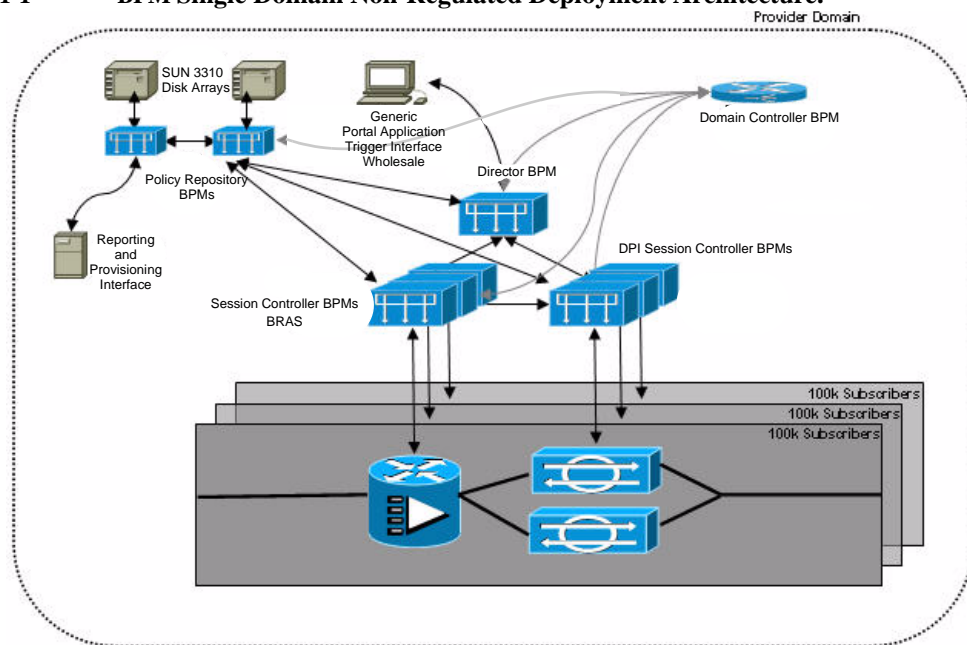
- [Architecture](#)
- [NP SP Deployment Detail](#)
- [Data Structure and Repository](#)

## Architecture

### Nonregulated Deployment

The BPM architecture in [Figure 1-1](#) depicts a non-regulated deployment.

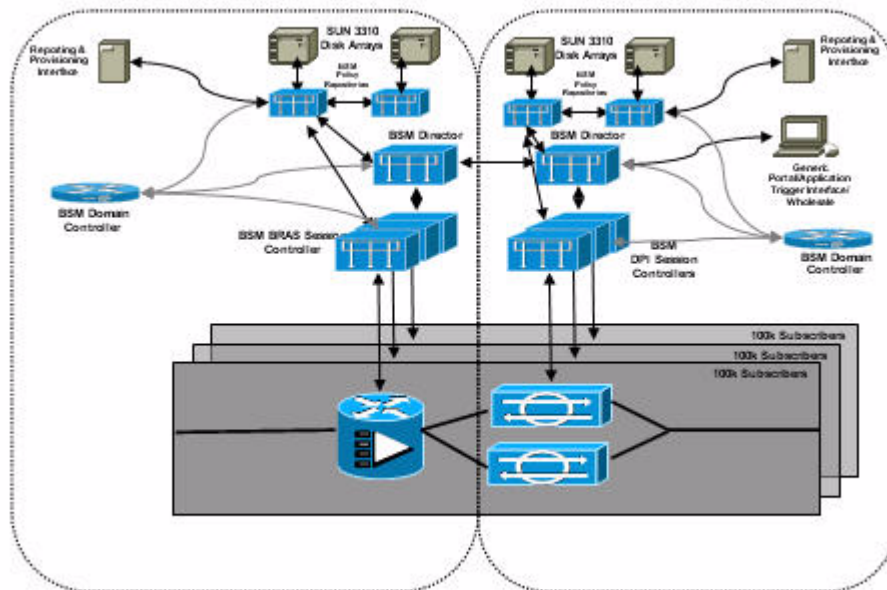
Figure 1-1 BPM Single Domain Non-Regulated Deployment Architecture.



## Regulated Deployment

The BPM solution is designed with integrated wholesale support and supports a regulated deployment.

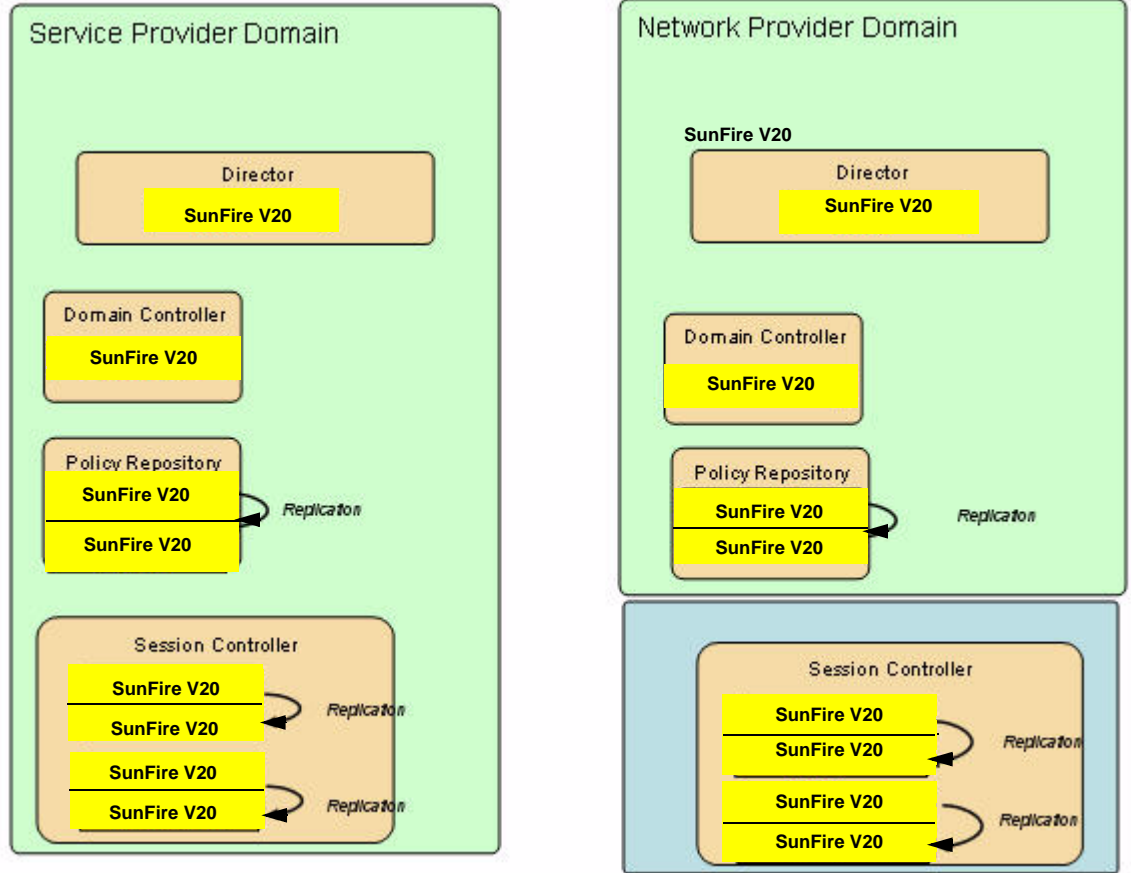
Figure 1-2 Regulated Deployment Architecture



# NP SP Deployment Detail

Figure 1-3 depicts the environment for deploying the Cisco Broadband Policy Manager (BPM) for the BPM application into an existing network. For each additional 100,000 subscribers, you must add an addition Session Controller BPM pair.

Figure 1-3 SP and NP Domain Deployment Details



## Failover

In an environment with active-standby clustering, if a primary BPM node fails, a secondary BPM system in the same data center becomes the *active* system. In a data center failure, the cold-standby becomes active. It does not have the state for sessions held in the failed data center. Spreading active clusters over two data centers reduces by half the number of required cold-standby devices.

## Data Structure and Repository

This section discusses the Cisco data structure and repository that store information for the Cisco Secure Access Application. It also provides Broadband Policy Design Studio (BPDS) screens for creation, display, updating, and deleting data objects stored in the data repository.

## System Support Type

You can configure a Cisco Broadband Policy Manager (BPM) as one of five system support types:

- Network
- Service
- Combined
- Service Director
- Network Director

## Network Support

A BPM configured for network support uses data objects pertaining to the configuration and operation of the network (device types, devices, device rules).

## Service Support

A BPM configured for service support uses data objects pertaining to subscribers and the services they use (domains, service groups, subscribers, services).

## Combined Support

A BPM configured for combined support uses data objects found in a network configuration and data objects found in a service configuration.

## Service or Network Director Support

A BPM configured for director support uses objects that provide the information required to route requests to other BPMs.

## Setting System Type

Configure a BPM device by importing the *secureAccessDatabaseGUISupport.tzz* file. Once you import this file, configure the system type by creating a BPM configuration setting the system type (Network, Service, Combined, Network Director, or Service Director).

Before the metadata is loaded, the options under the *New* option from the *File* menu or from the navigator tree allow for the creation of a rule or a policy function. Once the metadata is loaded, this option is expanded for the creation of a BPM configuration.

Add the BPM configuration by selecting the *New* option from the *File* menu or by importing a policy data file that contains the BPM configuration. However, do not create a BPM configuration using the *New* option from the *File* menu for a system that you plan to import policy data onto.

If the imported data contains a BPM configuration, the BPDS detects this and adjusts accordingly. If the imported data file does not contain a BPM configuration, the imported data does not appear in the navigator tree until a BPM configuration is added by selecting the *New* option on the *File* menu.

These configurations can contain rules and policy functions that determine the interaction of the data objects in each configuration. These configurations (except for the director configuration) can also schedule and grant access to services on a specific BPM. A BPM configured for network support can schedule and grant access to services only for administrators. A BPM configured for service or combined support can schedule and grant access to services for subscribers, as well.

## Metadata

In the GUI Object Manager tool, metadata represents the data structure. You can import metadata to invoke a structure for your database. The metadata file *secureAccessDatabaseGUI.tzz* exists for all PCS types. You should load it on every BPM system.

This metadata requires the type be set in the BPM configuration to determine the system type:

- Network BPM
- Service BPM
- Policy Repository BPM
- SP Director BPM
- NP Director BPM

On a new BPM, you can import the configuration with the other policy data or create it via the BVDS on a system in which the data is not being imported. If you import the data for a system that does not contain an BPM configuration, you should import the data first.

## WSDL Files

WEB Service Description Language

The SOAP driver contains a pin which that specifies the operation to invoke from the WEB Service Description Language (WSDL) file, if more than one operation is available. If only one operation is listed in the WSDL file, the SOAP driver uses that.

## Common Architecture

Network, service, and combined BPM configurations share the same underlying service scheduling and a common architecture. The architecture allows for scheduling of, and access to, services on a BPM with any configuration type. The common architecture also provides rules and policy functions to control access to network devices and services.

## Service Access

A BPM with a network configuration allows access to services and scheduling of services for the administrator only. A BPM with a service or combined configuration also allows access to services and the scheduling of services for subscribers. A service in the BPDS Subscriber Access view represents a Cisco flow. If the Cisco flow processes device rules, you can assign a profile to the service to provide those device rules.

## Service Event

A service can be run on demand by a subscriber, or it can be scheduled to run for a subscriber. For all BPM configuration types, a service event determines how and when a service is run by, or for, a subscriber. When scheduled to be run for a subscriber, the service can run immediately upon subscriber log on, at a period of time after logon, or at another, specific time. If the service uses network events (device rules), a relative or absolute end time can be specified. This can denote that a service continues for a specific period of time or terminates at a specific time. A service event can schedule regular service execution, based on an incremental time schedule, specified in seconds, minutes, hours, days, weeks, months, or years.

Services are scheduled in service events. The service events are aggregated into service profiles. A service event specifies whether a service is run directly by the subscriber on-demand or is scheduled for the subscriber at logon. A scheduled service event is programmed into the BPM system scheduler, based on the parameters defined in the service event.

Parameters can include execution immediately, after a period of time, or at a specific time. The parameters can include execution to completion, for a period of time, or until a specific time. Optionally, a service event can schedule a service to execute on a recurring schedule, based on seconds, minutes, hours, days, weeks, months, or years.

## Service Event Quota

On a BPM configured as a service system, quotas can be attached to a service event when the service profile that owns the service event is a bundle. In addition to volume (upstream or send, downstream or receive, or total), service event quotas can limit the time of service use and its number of allowed uses within a specific time period. Service event quotas define a service to be invoked for a subscriber, if he exceeds the quota limit. Service event quotas can define a profile to override the preconfigured service profile. Service event quotas can also receive supplements.

## Service Profile

A service profile is a collection of service events or a collection of services and information about how each service is executed. A set of service events can be packaged into a service profile. Service profiles associated with administrators, subscribers, service groups, and domains grant access to these service collections. A service profile used alone or with other service profiles can define the total set of services available to a service group. Service profiles can also be assigned directly to a subscriber or to the domain that contains the subscriber and the service group.

On a BPM with a network configuration, service profiles associated with the administrator grant him access to services in the service profile, or profiles, and service scheduling. In the network configuration, service bundles and the concept of usage quotas do not exist. On a BPM configured as a service system, one or more service profiles associated with a service group represent the set of services available to a subscriber in this service group and how a group subscriber can gain access to these services.

A domain can also have a set of service profiles. This helps determine the set of allowed services when only the domain is known or when neither the subscriber nor the service group has an associated service profile. Also, a set of service profiles can be associated with a subscriber to override the service profiles associated with the subscriber service group.

A service profile can be used as a service bundle. In addition to grouping the services that a subscriber has access to (either directly or through his service group or domain), a service bundle can have quotas that set volume limits for any subscriber who uses the bundle. Each quota can have one or more supplements that allow the quota to be extended. Quotas and supplements can have usage threshold values.



## Flow

For all BPM configurations, a service provides access to a Cisco flow. Upon selecting a flow, the name of the service is set and the trigger type of the flow is marked (socket, HTTP, event). If the Cisco flow can process device rules, a profile can be associated with the service to provide a sequenced set of device rules.

## Service Profile Quota

On a BPM configured as a service or combined system, a quota limits the allotted bandwidth volume (upstream or send, downstream or receive, and total) and defines a breach service for activation if the subscriber exceeds his quota limits. A quota can also specify a network profile for use by the breach service, overriding any other profile configured for use.

A quota can use supplements that allow a subscriber to increase the quota limit. A quota can also use usage thresholds. When a quota threshold is set, an event can notify an external application that a subscriber reached the threshold.

## Quota Supplement

On a BPM configured as a service system, quota supplements define standard additions to a service profile or service event quota and their charges. Quota supplements are created separately from the quotas to which they apply, since one supplement can be used by many quotas.

A subscriber can purchase supplements as long as the resulting quota is less than the supplement limit of the service profile (if the service profile defines a supplement limit). A supplement can send an event when the threshold is reached.

## Usage Threshold

A usage threshold defines a number (usually in bytes) to indicate when an external process is warned that a subscriber is approaching his quota. On a BPM configured as a service system, an event interface can be selected in the BPM configuration. When the event interface is set to JMS or SOAP, the appropriate protocol issues an event. For example, for JMS, it is issued to a registered application defined within the Service BPM to an external process. The external process can act, based on the event information.

A quota threshold value is measured against the quota. A supplement threshold is measured against the supplement. The threshold measurement panel allows for their modification.

When creating a new service profile, the default behavior is in effect, and both the quota and the supplement check boxes are selected. Threshold values can be measured against the sum of the quota and its supplements by checking the aggregate check box and adding a usage threshold value to aggregate thresholds list. Either the quota checkbox or the supplement check box can also be selected. Here, only thresholds on that object are measured against it.

You can add, update, and remove quotas using the quotas panel. (You must save the service profile to the database as a bundle before you can add quotas.)

The supplement limit sets the maximum amount that the quota(s) in the bundle can be increased using quota supplements. Once the bundle is fully configured, activate it by placing a check mark in the bundle activated check box and press the Submit button. If this is successful, the red DISABLED label becomes a green ENABLED label. Once a bundle is enabled, it cannot be modified. Until a bundle is enabled, subscribers who use this bundle cannot log onto the network.

## Administrator

All BPM configurations can have an administrator. A BPM configured for use as a network system has an administrator for the entire system. A BPM configured as a service system can have an administrator for each domain. A domain administrator is created and updated on the domain screen for a specific domain. An administrator can have service profile sets that schedule service execution execute and define administrative services for execution on demand.

## BPM Role Definition

The BPM configuration defines the role of a BPM and locates its resources. Some resources, such as the local communication port, are found on the BPM itself. Other resources, such as remote data stores and network, service, and repository BPMs can point to other Cisco BPMs.

The BPM configuration type entry must agree with the application (services and agents) on the target BPM. The type signals the BPDS to provide a view into the database tables that is consistent with the BPM system type.

With regard to the network BPM entry and the service BPM entry on the BPM configuration, the host and port must contain the host and port of the BPM that is handling the specific request type for this BPM.

## Network

If this BPM has a network configuration, the host and port of the network BPM in the BPM configuration contain the host name (or IP address) and port of this BPM itself. The service BPM host and port contain the host name (or IP address) and port of another BPM with a service configuration that handles service requests for this BPM.

## Service

If a BPM has a service configuration, the host and port of the service BPM in the BPM configuration contain the host name (or IP address) and port of this BPM itself. The network BPM host and port contain the host name (or IP address) and port of another BPM with a network configuration that handles network requests for this BPM.

## Combined

On a BPM with a combined configuration, both the service BPM and the network BPM entries in the BPM configuration contain the host name (or IP address) and port of the BPM itself.

The URL field for both the network BPM and the service BPM represents the name (servicename/flowname) of the service on a BPM that processes requests from other BPMs. The default URL for a service BPM is *processNtwkEvent/processNtwkEvent*. The default URL for a network BPM is *processServiceEvent/processServiceEvent*.

When the BPMs in a system use these defaults, the URL fields can be left blank. If you rename these services, you must enter the name of the new URL here. You must also ensure that all arguments to these services remain the same as well. (You can change only the name of the service/flow, if these services are to function correctly).

If a BPM with a director configuration handles requests for more than one BPM, the URL is:

Service BPM URL:

*processServiceEventDirector/processServiceEventDirector*

Network BPM URL:

*processNtwkEventDirector/processNtwkEventDirector*

As a repository, the BPM acts as a central data store from which other BPMs can cache a subset of the repository data. A repository itself does not act as a network or service BPM nor does a repository use another repository BPM, so these are not configurable once the repository check box is selected. A repository can have remote data stores so the remote data stores table remains enabled.

## Remote Data Store

A BPM can use distributed data stores (other BPMs) to store and retrieve parts of its overall data. A BPM can also hold a subset of records held on a larger repository. The remote data store entry specifies the host and port of the remote data store and the type of data stored there (administrator, device, policy, or session).

The application (the rules and services) on this BPM uses this remote store to save and retrieve the type of data. When the data store specified is the policy data store, the application database connection must be reset for the BPDS.

To reset the application database connection, a username and password must be supplied for access to the remote data store. Once authentication is complete, data from the remote store is displayed in the Subscriber Access view. Metadata can be loaded and objects can be added, deleted, and updated.

Since the BPDS continues to use the event bus on the local BPM, the application running on the BPM and the BPDS are made aware of changes on the remote store. When a change is to be made in the BPM configuration, the application database connection must be reset to that of the local BPM.

While connected to the application database on the local BPM, only BPM configuration should be changed when a remote policy database is used. When this is changed, the application database connection should be reset to point to the remote data store.

## Data Type

The policy data breakout can add three data types: administrators, device types, and devices. You can specify these data types separately from the policy database. This is useful when the policy database (or data store) uses the Policy Repository BPM, and the administrator and device information can be stored on a local data store.



### Note

If the device type is used on a Policy Repository BPM for a common set of device rules, the model, vendor, hardware revision, and software revision **MUST** match exactly on both platforms.

For configuration help, refer to the document in the References Section entitled Cisco Secure Access Database and Broadband Policy Design Studio (BPDS) Support. The following example shows how to break out session and policy data but applies to the other tables as well.

## Session and Policy Database on Local BPM

When both the session and policy configuration data are on the local BPM being configured, no entry is necessary. The lack of an entry causes a default to the local BPM device for the database.

## Session Database on Remote BPM / Policy Database on Local BPM

When the policy database is on the local BPM, no entry is necessary. The lack of an entry causes a default to the local BPM. When the session database is on a remote BPM, create a remote data store entry for the session database (name: session), containing the IP address (hostname) and the base port for the remote BPM.

## Policy Database on Remote BPM / Session Database on Remote BPM

When the session database is on the local BPM, no entry is necessary. The lack of an entry causes a default to the local BPM. When the policy database is on a remote BPM, create a remote data store entry for the policy database (name: policy), containing the IP address (hostname) and the base port for the remote BPM.

## Session and Policy Database on Remote BPM

When the Session database is on a remote BPM, create a remote data store entry for the session database (name: session), containing the IP address (hostname) and the base port for the remote BPM. When the Policy database is on a remote BPM, create a remote data store entry for the policy database (name: policy), containing the IP address (hostname) and the base port for the remote BPM.

## Remote Database Store Package Requirements

When creating a policy database remote store, load the following .tzz packages:

- metadata (combined, network, or service)
- policy data (combined, network, or service)

When creating a session database remote store, load the following .tzz package:

- metadata (combined, network, or service)



**Note**

---

The remote data store functionality works for both the network and service BPMs. However, current installations utilize Network BPM remote data stores only.

---

## Timeout

Timeouts define the time allotted for request completion. The system defines the following three timeout types:

Inter-BPM (*interBPMCommunicationTimeout*)

Intra-BPM (*intraBPMCommunicationTimeout*)

Session State (*sessionStateTimeout*)

These define the maximum time allotted for requests to complete between two BPMs, within the BPM itself, and when changing the state of a session. Upon a *sessionStateTimeout*, the type of session state can be set and a service invoked via the socket agent can be selected. When a session remains in the selected state beyond the specified number of seconds, the selected service is invoked.

## Rule and Policy Function

All BPM configurations contain rules and policy functions. Rules and policy functions can be used in Cisco flows to represent policies that further control access to network resources. Rules are scripts that encapsulate data gathering, comparison, and calculation. Rules can invoke other rules directly by calling them as procedures. Rules can also become policy functions. Policy functions can be used in Cisco flows to invoke other agent functions or to trigger other policies or service flows.

### Rule

Rules are Tcl scripts that represent a single Tcl procedure. One rule can call another rule since the rule is itself a procedure. As you write the procedure, the system tracks the rule name, defined input or output parameters, and rules called from this rule (dependencies). Higher level rules can call on lower level rules that provide simple data gathering, comparisons, or calculations to carry out this work. However, higher level rules often represent functionality required in flows to carry out subscriber access policies. The higher level rules can become policy functions.

### Policy Function

Policy functions are rules encapsulated in a usage of the *executeScript* function of the tcl agent. A rule can be used in a Cisco flow to affect the way network access is allowed using that flow. For backward compatibility, a policy function can be scripted directly (with or without calling one or more rules).

## Network Configuration

Network configuration concerns network devices and the rules that control these devices to provide secure access to network services. Defined device types allow the grouping of network devices by attributes, such as vendor, model, and hardware and software revisions. Vendor and model are selected from predefined, allowed values.

### Device

A device represents an actual network device. A device is created by entering its name, IP address, and (optionally) MAC address. Also a device type selection identifies the device type or class. The secondary IP address is used if this device can fail over to another device.

If the access address is filled in, the access address programs the device when the device IP address is requested. Access address can also have a secondary address.

The keep alive limit defines the minimum number of pings that must succeed to consider this device usable. If the success rate is below this level, this device is unusable.

One or more management protocols must also be defined for a device to indicate the protocol(s) that can communicate with it. This allows for finer grained selection of device scripts within a device rule and ensures that a selected device script can be applied to a specific device.

## Device Type

A device type is a classification for actual devices. By specifying the vendor, model, and (optionally) the hardware revision and software revision, a class or group of devices can be defined. This grouping allows a device rule to be applied to a specific device. A device specific script is selected from a device rule, based on the device type associated with both the device and the device script.

## Device Rule

A device rule defines a general or abstract action performed on a device to carry out a network action. They represent the generic tasks applied to any device that supports the device rule functionality.

A device rule contains one or more device-specific scripts. Each script carries out the device rule functionality for a specific device type or group of device types. A script can carry out the abstract action defined by the device rule on a specific type or class of device using a specific management protocol. The management protocol on one router may require a different script from that of a router using a different management protocol.

A device rule can apply to a specific device by matching the device type of the device with the device type(s) of the device script. This optional type attribute creates a grouping for specialized rules.

Device rules are paired into install and uninstall rules within network events. The network events are sequenced into profiles that can set up and take down a subscriber session on the network. The install rules represent the device rules that set up a session for a subscriber when he logs on to the network. The uninstall rules represent the device rules that take down a session when a subscriber logs off of the network.

## Device Script

A device script belongs to a specific device rule. The device script carries out the action of the device rule on a device type or set of device types using a specific management protocol. The device rule has a name for easy identification. A type (or management protocol) is selected, based on the script. The script is entered in the script window and any device type that can use this script is added in the device types list. This information can locate the correct script to carry out a specific device rule, given a specific device, based on the device type of the device and the management protocols the device supports.

## Allowed Value

New vendors and models can be added to the allowed value table for selection on this screen. The allowed value table builds a list of accepted values for a simple type. Adding values to the model or vendor list allows these values to be displayed in selection lists on the device type screen.

## Management Protocol

You can define two management protocols for a device:

- CLI
- RADIUS Attribute

## Profile

A profile sets up and takes down a network session for a subscriber when the subscriber logs on, logs off, or changes his bandwidth. It does this through a series of sequenced network events. Once defined, a profile can be referenced by a service group, subscriber, domain, or service on a BPM configured for service support. A service that represents a Cisco flow can be associated with a profile. When this service runs, it uses the device rules defined in the profile to carry out its functionality. A rule can determine profile priority.

Profiles are also referenced on the network BPM itself by access lines and access groups. A service that represents a Cisco flow can also be associated with a profile. When this service runs, it uses the device rules defined in the profile to carry out its functionality.

An access line represents a customer of the network provider. The access line either has a default profile or inherits its default profile from its access group.

## Network Event

A network event contains an install device rule and a related uninstall device rule. The install rules create a network session, in step order. The uninstall rules take down this session, in reverse step order.

## Profile Proxy



Note

---

Profiles do not directly relate to other objects. They relate to other objects (service group, domain, subscriber) by a profile proxy.

---

A profile proxy contains only the id and name of the corresponding profile. In this way, objects on one BPM can reference profiles on another BPM. For example, a BPM with a service configuration does not contain actual profiles. Domains, service groups and subscribers on that BPM must reference profiles on a BPM with a network configuration. The network BPM sets up and takes down the network sessions.

To do this, the service BPM must contain profile proxies that represent the actual profiles on the network BPM. Network BPMs and BPMs with combined configurations also use profile proxies to relate profiles to other objects. On a network BPM, profile proxies always represent profiles on the network BPM itself. If a combined BPM is used as a stand-alone BPM (usually a demonstration box), all of the profile proxies represent profiles on that combined BPM.

## Access Line

An access line represents a customer of the network provider. Access lines can have a default network profile or inherit a default network profile from an access group. When establishing a network session, this profile is used unless an alternative profile is provided.

## Access Group

An access group is a collection of access lines that have similar network requirements and thus use the same default profile when establishing a network session. When access lines are created or updated, they can be placed in an access group. When one of these access lines establishes a network session, the network profile for the access group is used, unless the access line has its own profile or an alternative profile is provided.

## Service Configuration

Service configuration deals with subscribers, the network services they use, and their access methods. One or more service profiles (or a single service profile representing a service bundle) can be associated with a service group to represent the set of services available to a subscriber in this service group and the way that a group subscriber can gain access to the services.

A domain can also have a set of service profiles (or a single service profile representing a service bundle) associated with it. This can determine the set of allowed services when only the domain is known or when neither the subscriber nor the service group has an associated service profile. A set of service profiles (or a single service profile representing a service bundle) can be associated with a subscriber to override the service profiles associated with the subscriber service group. Usage of bandwidth volume, time, and service invocations is collected in usage records attached to each subscriber.

Collected usage statistics record the volume of traffic by type. This usage data, compared to quota and usage threshold data, can trigger a breach service, in case of a breach or notify an external process when a usage threshold is reached.

A subscriber can have subaccounts. A subaccount is a subscriber that cannot make a connection to the network itself but relies on the connection established by the master account for connectivity to the network. The usage and usage statistics information for a subscriber can be viewed from the subscriber screen. A subaccount inherits its domain and service group associations from the master account. A subaccount can have its own set of service profiles, which can only be a subset of the service profiles available to the primary account. The primary account can have its own set of service profiles or can inherit them from the primary account service group or domain.

Events can be sent to an external process using a SOAP interface (usually, another Cisco BPM) or a JMS interface (usually, a third-party registered application). A registered application is an external process that is to receive notification of events with in the Service BPM. These events are triggered when subscriber usage reaches a usage threshold.

## Domain

Domain represents a network domain and contains subscribers and service groups. A domain can have a network profile and/or a service profile. The network profile and system profile are used for a subscriber without a profile who belongs to a service group without profiles. The domain profile(s) are also used when no subscriber information is maintained on the BPM.

## Service Group

A service group contains subscribers who use the same type of services. The network profile and service profile for a subscriber comes from his service group.

## Subscriber

A subscriber can have his own profile and/or service profile(s) that can override the profile or service profile(s) associated with his service group. If he has no profile, his service group profile is used when establishing his network connection. If a subscriber has no associated service profiles, the service profile set of his service group is used. If his service group does not have an associated profile or the service group has no associated service profiles, any missing values are taken from the subscriber domain.



A subscriber can also represent a primary account that can contain subaccounts (other subscribers). This allows a subaccount member to use the subscriber network connection with his own subscriber name. Each subaccount can have its own service profiles. The set of service profiles available to a subaccount is limited to a subset of the service profiles available to the primary account. Before a subaccount can be added to a subscriber, the subscriber must be saved in the database as a primary account.

Add subaccounts by right-clicking the subaccount table and selecting the New Sub Account option. When adding one or more subaccounts a primary account, select one of the subaccounts to be the primary subaccount by selecting the primary account checkbox on the subaccount. The primary subaccount can add additional subaccounts to the primary account and can select the set of service profiles available to a subaccount. A subaccount can have a set of service profile(s) that override the service profiles on the primary account.

## Registered Application

A registered application can receive events using a JMS interface from the Service BPM. These are events that are triggered when a subscriber usage reaches a predefined usage threshold. The structure and content of the registered application events vary from application to application. The application designer and the owner of the external process must agree upon structure and content.

## Director Configuration

A BPM with a Director configuration routes requests to other BPMs. Subscribers (on the service side) and access lines (on the network side) can be distributed among several BPMs. When the director receives a request, it locates the BPM with the information for a subscriber or access line and forwards the request to that BPM.

### SP Director Entry

The SP Director entry on the service side represents subscriber. The routing index translates a request that is based on a network-side routing index into a request that is based on a username and domain on the service side.

### NP Director Entry

The NP Director entry on the network side identifies the NAS Address an access line uses, based on a routing index. The NAS Address finds the service BPM and network BPM that hold session information for this line.

### NAS Director Entry

The NAS Director entry used on the network side maps a NAS address to a set of network and service BPMs. Information about the access lines using this NAS address to enter the network, is found on the specified BPM(s).





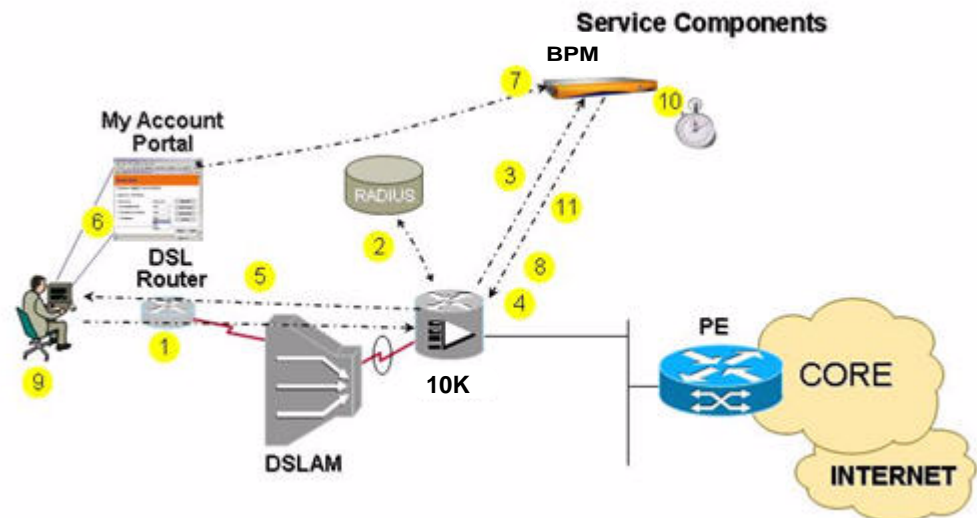
# Applications in Action

## Overview

This chapter discusses the following Broadband IP Services Module (BSM) scenarios:

- [BPM and 10K Applications](#)
- [BPM, 10K, and SCE 1010](#)

Figure 1-1 Sample BSM Application - Time-Based Turbo Bandwidth.



## BPM and 10K Applications

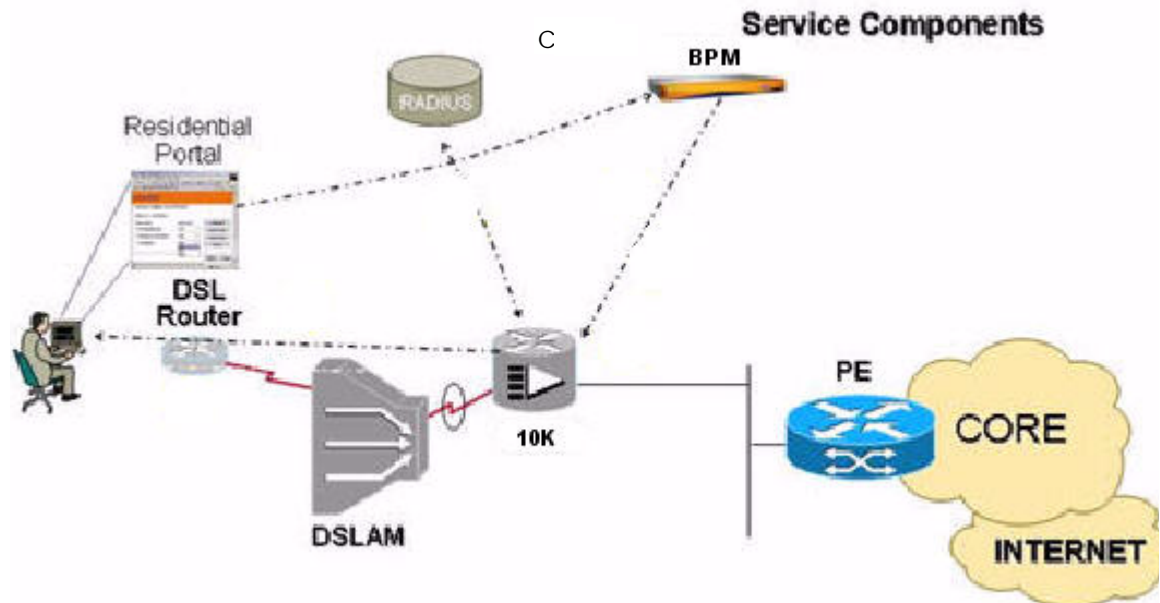
This section discusses the following applications:

- [Bandwidth on Demand](#)
- [Time-Based Turbo Bandwidth](#)

## Bandwidth on Demand

The Bandwidth-on-Demand (BoD) application, a dynamic subscriber-initiated service, demonstrates self-provisioning and subscriber control of the available current session bandwidth. In the BOD application, the subscriber can dynamically change his Internet bandwidth, without intervention by the service provider. The service provider can offer additional revenue-producing services beyond the standard offerings. A subscriber usually receives a default, a standard bandwidth setting that is sufficient for most applications. However, if he requires additional bandwidth, for example, to download a large file, he can dynamically change his bandwidth. This occurs without action by the service provider.

Figure 1-2 Bandwidth-on-Demand.



### How It Works

The BPM device presents dynamic service catalog information to the subscriber, who can select his own service level. The BPM device interfaces to other applications to facilitate dynamic network changes.

1. The subscriber logs onto the network via a PPP client over a tethered Ethernet connection.
2. The PPP session is terminated at the 10K Broadband Remote Access Server (10K BRAS device) device.
3. The BPM device receives the accounting start and determines the subscriber network profile by accessing the policy database.
4. The BPM device configures the session with the appropriate service level bandwidth and authorizes the subscriber.
5. The subscriber receives authentication and his default service level bandwidth. He can utilize network resources.
6. If the subscriber wants a higher service level bandwidth, he brings up the control panel and selects the higher service level. This triggers the services profile change held in the policy database and results in a dynamic configuration of the network device with the new service level bandwidth.
7. The subscriber receives his new service level bandwidth.

## BPM Functions

The BPM device handles the following functions:

- Determines the default service level policy from subscriber information.
- Handles policy-driven dynamic configuration of the network, based on subscriber session parameters.
- APIs provide service exposure to external applications for change in policies and services.
- 10K BRAS Device Interaction:
  - Policing
  - Marking, for per-application boost

## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- RADIUS
- SNMP

## Base Capabilities

- Subscriber authentication
- Start/Stop accounting for turbo mode
- Graphical/tabular administration functionality to configure turbo policy value (that is, 128kb/s normal: 256kb/s turbo)
- Offline interface to look up CAC for permit or denial of the request. If another system captures this information and the BPM device determines permit or deny, based on a simple rule. This rule states if the current aggregate bandwidth plus the request increment 256 Kbps is greater than the total aggregate, deny the request.
- Wholesale SOAP interface to trigger service

## Application Benefits

This application provides the following:

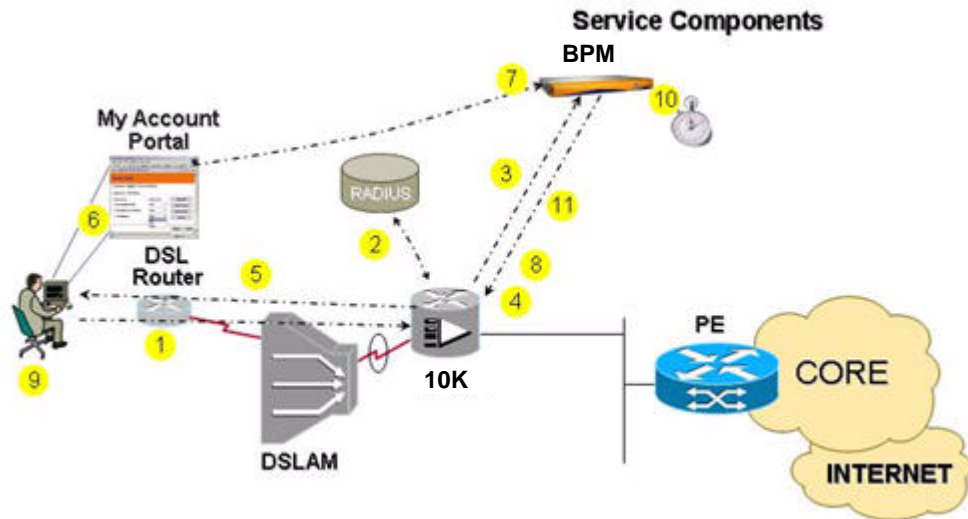
- Additional value for basic RADIUS accounting, adding rich service profile and fine-grained policy information
- Dynamic use of network policies in service profiles for configuration of the related session in real time
- Subscriber interface to facilitate network changes
- Presentation of dynamic service catalog information to subscriber for service selection
- Support of timed service events via integral scheduler

## Time-Based Turbo Bandwidth

The Time-based Turbo Bandwidth application operates directly on the customer experience and is driven by the subscriber contract. Time-based services have similar status issues to usage- or quota-based options. Both operate directly on the customer experience and are driven by the user contract context.

During low system-usage periods, the administrator can extend or vary subscriber or supplier contract terms, so that spare capacity can earn, rather than remain idle. The concepts of an online daytime use by subscribers and interactive sessions, followed by an online nighttime use for system backups (with a potential for service increment) may be attractive to service providers.

Figure 1-3 Time-Based Turbo Bandwidth.



## How It Works

The scenario below depicts the ability of the BPM device to support timed service events using an integral scheduler.

1. The subscriber logs onto the network via a PPP client over a tethered Ethernet connection.
2. The PPP session is terminated at the 10K BRAS device.
3. The BPM device receives the accounting start and determines the subscriber network profile by accessing the policy database.
4. The BPM device configures the session with the appropriate service level bandwidth.
5. The subscriber receives authentication and receives his standard service level bandwidth. He can now utilize network resources.
6. If the subscriber wants a service that requires a higher bandwidth, such as watching video clips or downloading applications, he selects a Turbo bandwidth service for a specific period of time.
7. The BPM device registers the change of service request from the subscriber and updates the session information for the subscriber. It also dynamically sends the new session information to the 10K BRAS device. It sets up a time session event for the specified timer period of the Turbo service.
8. The BPM device dynamically reconfigures the 10K BRAS device for the new session bandwidth time period.
9. The subscriber receives his Turbo bandwidth for the selected period.
10. When the time period expires, the BPM device determines the standard policy for the subscriber and updates the real-time session information. It reconfigures the subscriber session on the 10K BRAS device.
11. The BPM device returns the 10K BRAS device session to the standard rate. The subscriber service level bandwidth returns to the standard rate.

## BPM Functions

The BPM device performs the following functions:

- Supplements basic RADIUS accounting, adding rich service profile and fine-grained policy information.
- Dynamically uses network policies from service profiles and configures the related session at the device level, in real time.
- Presentation of dynamic service catalog to subscriber for service selection
- Support for timed service events via an integral scheduler



## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- RADIUS
- SNMP

## Base Capabilities

- Subscriber authentication
- Start/Stop accounting for application selection
- Graphical/tabular administration for setting the turbo timer
- Graphical/tabular administration functionality to configure turbo policy value (that is, 128 Kbps normal; 256 Kbps turbo)
- Offline interface to look up CAC for permit or denial of the request. That is, if another system captures this information and the BPM device evaluates permit/deny, based on a simple rule. If the current aggregate bandwidth plus the request increment 256 Kbps is greater than the total aggregate, deny the request.
- Wholesale SOAP interface

## Application Benefits

This application provides the following:

- Interface to applications to facilitate dynamic network changes
- Presentation of dynamic service catalog to subscriber for service selection
- Support for timed service events via an integral scheduler

# BPM, 10K, and SCE 1010

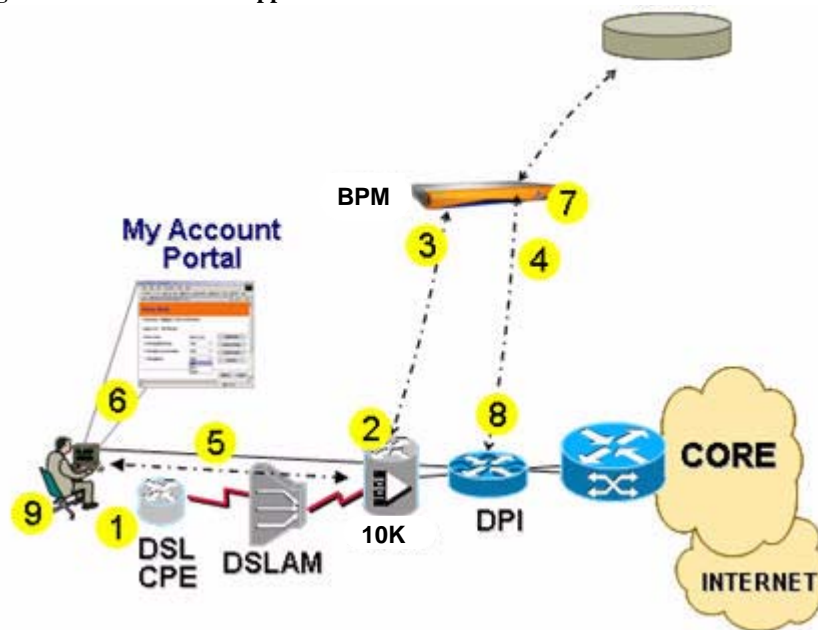
This section discusses the following BPM, 10K, and SCE 1010 applications:

- [Tiered Applications](#)
- [Quota-Based Tiered Applications](#)
- [VoIP-Oriented Preferential Applications](#)
- [Firewall Application](#)

## Tiered Applications

Tiered Applications give the service provider the ability to bundle applications and target them to specific subscriber bases. This service uses the familiar gold, silver, and bronze Internet service levels, but offers a new dimension by allowing the tiers to stratify availability and performance at an application level. This service identifies a subscriber and automatically deploys his default profile, the tiered application package. For example, the subscriber with a bronze tiered package receives email and Internet access at 512 Kbps, denying other traffic. The subscriber can modify his service by selecting an alternative package.

Figure 1-4 Tiered Applications.



## How It Works

1. The subscriber logs onto the network via a PPP client over a tethered Ethernet connection.
2. The PPP session is terminated at the 10K BRAS device.
3. The BPM device receives the accounting start and determines the subscriber network profile by accessing the policy database.
4. The BPM device associates the subscriber identity and deploys the corresponding policy to the Service Control Engine (SCE) SCE 1010.
5. The BPM device authorizes the subscriber, who receives his service level bandwidth and can utilize network resources.
6. The subscriber opts to use an application not permitted by his default package, for example, Voice-over-IP (VoIP).
7. The BPM device registers the change of service request from the subscriber and updates the subscriber session information.
8. The subscriber receives the new session bandwidth and can use VoIP.

## BPM Functions

The BPM device performs the following functions:

- Supplements basic RADIUS accounting, adding rich service profile and fine-grained policy information.
- Dynamically uses network policies from service profiles and configures the related session at the device level, in real time.
- Interface to applications to facilitate dynamic network changes
- Presentation of dynamic service catalog to subscriber for service selection

## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- P-Cube
- RADIUS
- SNMP

## Base Capabilities

- Graphic administration of service tier definition. Options include the following:
  - Default Access Bandwidth Value Up/Down Options
  - Default Applications (P2P, IM, VoIP, Internet, SMTP, etc.)
  - Default Application Bandwidth (Up/Down)
- Subscriber Authentication
- Subscriber Service Change
- Accounting for Tier Change
- Wholesale SOAP Interface

## Application Benefits

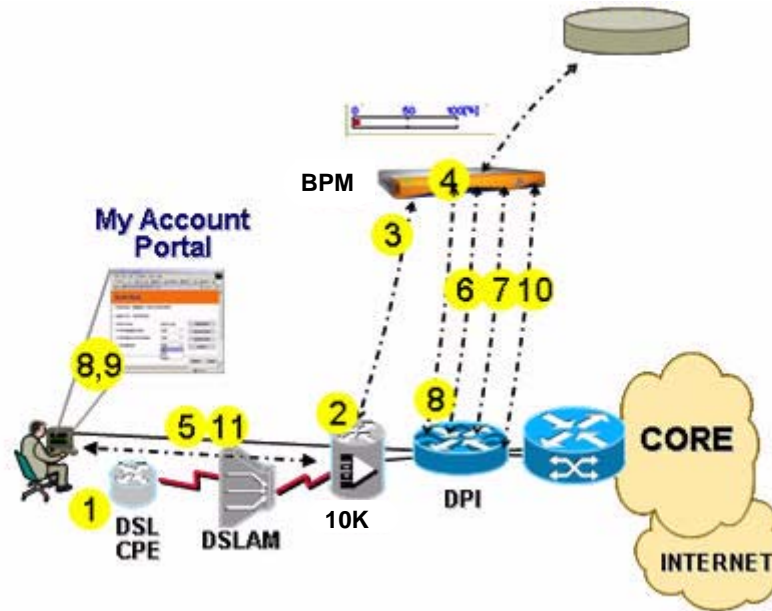
This application provides the following:

- Interface to applications for dynamic network changes
- Presentation of dynamic service catalog information to the subscriber
- Integration of traditional access methods with new technology, for example, the SCE 1010

## Quota-Based Tiered Applications

Quota-based tiered applications add another dimension to service tiers. Quotas can be applied to any application within the service tier and can be offered in the familiar gold, silver, and bronze levels. The subscriber can modify his service use or purchase additional quota. If the subscriber breaches his quota, the BPM can deploy controls that restrict or diminish the application performance on the SCE 1010. Or, the subscriber can purchase incremental quota until the quota refresh at the end of his service cycle, for example after 30 days.

Figure 1-5 Quota-Based Tiered Applications.



### How It Works

1. The subscriber logs onto the network via a PPP client over a tethered Ethernet connection.
2. The PPP session is terminated at the 10K BRAS device.
3. The BPM device receives the accounting start and determines the subscriber network profile by accessing the policy database.
4. The BPM device associates the subscriber identity, notes the remaining quotas of his applications, and deploys the corresponding policy to the SCE 1010 device.
5. The BPM device authorizes the subscriber. He receives his default service level package and can utilize network resources.
6. At regular intervals, the SCE 1010 solution set provides periodic updates via RDRs to the BPM device. The BPM device evaluates the remaining subscriber quota and determines whether or not to deploy a new policy.
7. The BPM device deploys any updated policy to the SCE 1010, denying particular application traffic. Alternatives, such as lowering bandwidth to 56k can also be applied.
8. If the subscriber attempts to access the blocked service, he receives a denial.
9. The subscriber selects a quota increment package.

10. The BPM device refreshes the quota counters and deploys the appropriate policy package to the SCE 1010.
11. The subscriber resumes service use.

## BPM Functions

The BPM device performs the following functions:

- Supplements basic RADIUS accounting, adding rich service profile and fine-grained policy information.
- Dynamically uses network policies from service profiles and configures the related session at the device level, in real time.
- Persists quota information across session state.
- Evaluates dynamic policy execution, based on quota triggers.

## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- P-Cube
- RADIUS
- SNMP

## Base Capabilities

- Graphic administration of service tier definition. Options include the following:
  - Default Access Bandwidth Value Up/Down Options
  - Default Applications (P2P, IM, VoIP, Internet, SMTP, etc.)
  - Default Application Bandwidth (Up/Down)
- Graphical tool that allows administrators to set quota refresh page values

- Graphical tool that configures policies for actions following a quota breach, that is, deny, send bandwidth to 56 Kbps
- Subscriber Authentication
- Subscriber Service Change
- Accounting for Tier Change
- Accounting for Quota Replenishment
- Administrative Broadband Policy Design Studio (BPDS) for quota management
  - Quota Reset Timer, that is, every 30 days, 7 days, etc.
  - Quota Assignment per Application, that is, 10GB for P2P
  - Downstream/Upstream/Aggregate checkbox, that is, 10GB for P2P Aggregate Usage
  - Quota Adjustment Tool (reset subscribers, reset subscriber groups, add/subtract quota from subscribers, add/subtract quota from subscriber groups)
- Purchase of incremental quota to supplement their monthly quota
- Wholesale SOAP Interface

## Application Benefits

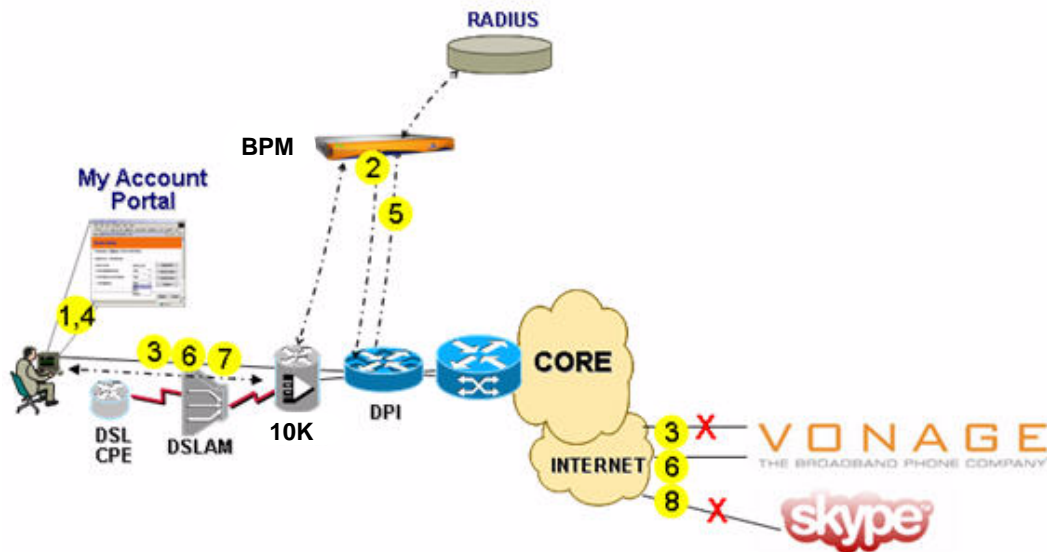
This application provides the following:

- Integration of new technology, for example, SCE 1010, with traditional access methods
- Differentiation of Layer 7 applications and application of preferential treatment to in-house derived applications
- Presentation of dynamic service catalog information to the subscriber for service selection
- Incorporation of quota-based logic to further control services and differentiate offerings

## VoIP-Oriented Preferential Applications

Using the SCE 1010, service providers gain new insight to the applications traversing their networks. With the BPM device and the SCE 1010 device, similar applications can be controlled, based on their unique signature. Based on the subscriber service package, the BPM device determines whether the subscriber has the *Basic Internet* or the *Internet with VoIP* package. If the subscriber subscribes to a Voice-over-IP (VoIP) package that the service provider sponsors (Vonage in Figure 1-6), he receives the *Internet with VoIP* package. This package provides Vonage calls with an improved service quality. The subscriber with the *Basic Internet* package does not receive Vonage access. Subscribers with either package cannot use VoIP applications that their service provider does not support.

Figure 1-6 Preferential Applications - VoIP-Oriented Service.



### How It Works

1. The subscriber selects the *Basic Internet* package.
2. The BPM device deploys the subscriber package to the SCE 1010 device.
3. The subscriber attempts a Vonage call and receives a denial.
4. The subscriber upgrades to the *Internet with VoIP* package.
5. The BPM device deploys the updated subscriber package to the SCE 1010.
6. The subscriber again attempts a Vonage call. He finds that the call does go through and that he has improved service.
7. The subscriber attempts to use an unapproved VoIP package, for example, Skype.
8. Per the Skype signature, the SCE 1010 denies the attempt.



## BPM Functions

The BPM device performs the following functions:

- Dynamically uses network policies from service profiles and configures the related session at the device level, in real time.
- Associates subscriber identity and deploys corresponding policy to the SCE 1010 device. The package can contain a mix of application access/deny and bandwidth settings.

## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- P-Cube
- RADIUS
- SNMP

## Base Capabilities

- Variation of Tiered Application
- Support for the definition of tiers that differentiate between similar applications and provide preferential treatment, based on the application signature
- Wholesale SOAP interface

## Application Benefits

This application provides the following:

- Interface to applications to facilitate dynamic network changes
- Presentation of dynamic service catalog information to the subscriber for service selection
- Integration of new technology, for example, SCE 1010, with traditional access methods

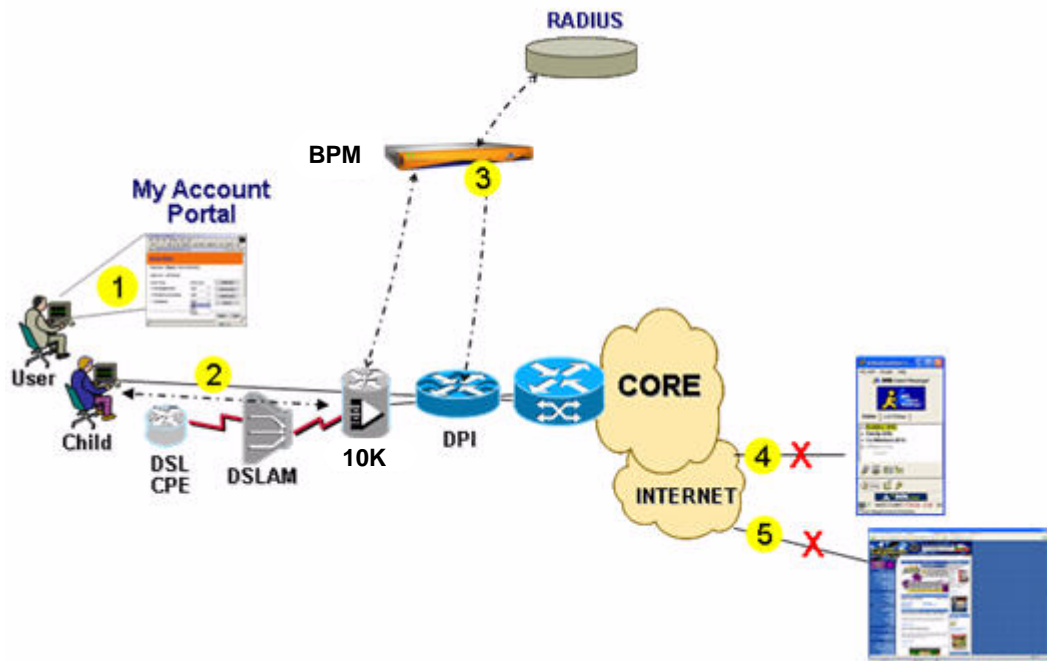
- Differentiation between differing VoIP packages, based on their unique signatures.

## Firewall Application

This family control-oriented application allows a subscriber to control access to services and web sites that can, and cannot, be accessed from his subscriber line. It allows him to tailor the available services to govern when and what his family can access. The scenario in Figure 1-7 focuses on parental control of access to services (VoIP, P2P, and IM) and web sites

A service provider can also use the application to control access to applications and web sites. The firewall application can be operated by the service provider control to limit access to particular services for groups.

Figure 1-7 Parental Control-Oriented Application.



## How It Works

1. The subscriber (parent) logs into the service control panel and selects the services to be made available for his subaccount (child). He can restrict the time of day that those services are to be available (for example 24 x 7 or a specific window). He can also restrict access to web sites. (This could be in addition to various filter tools.)
2. The child logs onto the network.
3. Based on parental preference, the BPM device deploys the corresponding package to the SCE 1010.
4. If the child attempts to use a blocked service, for example Instant Messaging, during a restricted time frame, for example from 11 p.m. to 7 a.m., he cannot.
5. If the child attempts to access a restricted web site during a restricted time frame, he cannot.

## BPM Functions

The BPM device performs the following functions:

- Exposes configurations to subscriber.
- Allows control of subaccounts.

## Access Methods

- PPPoA in PTA or LAC/LNS Mode
- PPPoE in PTA or LAC/LNS Mode
- PPPoEoA in PTA or LAC/LNS Mode
- PPPoA in L2TP Mode
- PPPoE in L2TP Mode
- PPPoEoA in L2TP Mode
- PPPoA in RA-MPLS Mode
- PPPoE in RA-MPLS Mode
- PPPoEoA in RA-MPLS Mode

## BPM Agents

- 10K CLI
- Application Trigger
- Generic SOAP Partner
- P-Cube
- RADIUS
- SNMP

## Base Capabilities

- Administration functionality to allow applications in a homeowner service tier to be subassigned, based on time of day/day of week

- Ability to enforce policy requests, based on subscriber login or line
- Wholesale SOAP interface to trigger service

## Application Benefits

This application provides the following:

- Integration of new technology, for example, SCE 1010, with traditional access methods
- Ability for subscriber to dynamically configure subaccounts
- Time-of-day policies to improve service offering value
- Real-time policy reconfigurations on application accessibility



# Application Subscriber Login and RADIUS Processing

## Overview

The NP BPM can act in any of the following roles in the RADIUS protocol exchange sequence:

- RADIUS Authentication Proxy (not currently supported)
- RADIUS Accounting Proxy
- RADIUS Accounting Server



**Note**

The NP BPM can act in any of the above roles. It must ensure that, through the various RADIUS phases (or phase) that it is part of, it obtains all of the required RADIUS information. If not, the subscriber does not receive access through the NP BPM.

## RADIUS Accounting Message Interception

The NP Session Controller BPM is responsible for intercepting Accounting Messages with the status type field set as follows:

- Accounting Start (1)
- Accounting Stop (2)
- Interim Accounting (3) (Not supported yet)
- Accounting On (7) (Not supported yet)
- Accounting Off (8) (Not supported yet)

The NP Session Controller BPM is responsible for obtaining following subscriber information:

- Username (attribute 1)
- Domain (attribute 1)
- Subscriber IP address (attribute 8)
- 10K BRAS device Address (attribute 4)
- 10K BRAS device Interface the subscriber logs in on (attribute 87/61, or Cisco VSA (26) depending on the setup)
- Accounting Session Id (attribute 44)

The NP Session Controller BPM (wholesale) does not have access to subscriber information, other than the 10K BRAS device and 10K BRAS port that the user dials in through. The 10K BRAS device address and a default network profile are configured for use by all subscribers when they initially log in. These are programmed in the Access Line Database. A default network profile exists for subscriber logon. If no profile is found for the 10K BRAS device the subscriber enters from, the Start Accounting message is rejected and no session is created.

All of the *real* subscriber information: username, domain, group, service profile/bundle and default network profile associated with the group, exists on the retail side.

The subscriber, prior to a complete retail/wholesale exchange, must not have a *floating* 10K BRAS device configuration. When the subscriber logs in, use of the 10K BRAS device address obtained from the RADIUS NAS-IpAddress locates a *temporary* network profile that limits his log in to *bare bones* access. The temporary network profile is in the access line entry. There is one access line entry per 10K BRAS device where the only elements looked at in the entry are the 10K BRAS device IP address and the network profile that is to be used as the *temporary* profile for that 10K BRAS device address. The following sections describe the NP Session Controller BPM actions upon receipt of the supported RADIUS messages.

It is important to understand the Cisco service breakout on the NP Session Controller BPM. The following services support RADIUS Message processing:

### Service - *genericRadiusStartAccounting*

- RADIUS Start Accounting
- Interim Accounting
- Accounting On messages

This service processes the RADIUS messages, asynchronously invoking the NP Session Controller state machine and responding to the RADIUS Accounting Request.

|              |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| Trigger Type | RADIUS                                                                                       |
| Inputs       | <i>accountingStatus</i><br><i>requestAttributes</i><br><i>responseAttributes</i>             |
| Outputs      | <i>responseType</i> for the RADIUS agent to return to the 10K BRAS device (Accept or Reject) |

### Service - *genericRadiusStopAccounting*

- RADIUS Stop Accounting
- Accounting Off.

This service processes the RADIUS messages, asynchronously invoking the NP Session Controller state machine and responding to the RADIUS Accounting Request.

|              |        |
|--------------|--------|
| Trigger Type | RADIUS |
|--------------|--------|

|         |                                                                                              |
|---------|----------------------------------------------------------------------------------------------|
| Inputs  | <i>accountingStatus</i><br><i>requestAttributes</i><br><i>responseAttributes</i>             |
| Outputs | <i>responseType</i> for the RADIUS agent to return to the 10K BRAS device (Accept or Reject) |

### Service - *genericNetworkStateMachine*

This service locates the temporary session record and takes the appropriate action, based on the event name and the session state.

|              |                                                      |
|--------------|------------------------------------------------------|
| Trigger Type | Socket                                               |
| Inputs       | <i>eventAction</i><br><i>temporary record handle</i> |
| Outputs      | result (ok or error string)                          |

### Service - *deviceConfig*

This service is responsible for programming the 10K BRAS device through a telnet connection. This service is invoked through the scheduler by the Cisco application creating a series of *now* (for immediate sequential execution) jobs that are handed off to the scheduler.

|              |                                                                          |
|--------------|--------------------------------------------------------------------------|
| Trigger Type | Socket                                                                   |
| Inputs       | <i>configScript</i><br><i>DeviceAccessInfo</i><br><i>InputParameters</i> |
| Outputs      | result (ok or error string)                                              |

### Service - *deviceConfigISG*

This service is responsible for programming the 10K BRAS device through an *ISG* interface using RADIUS VSAs. This service is invoked through the scheduler by the Cisco application creating a series of *now* jobs that are handed off to the scheduler.

|              |                                                                          |
|--------------|--------------------------------------------------------------------------|
| Trigger Type | Socket                                                                   |
| Inputs       | <i>configScript</i><br><i>DeviceAccessInfo</i><br><i>InputParameters</i> |
| Outputs      | result (ok or error string)                                              |

## RADIUS Accounting Proxy/Accounting Server Start Accounting Processing

- Step 1** The Accounting Session Id attribute determines if a session entry is in process for this subscriber. Both the temporary table and the database are checked for an existing session for this Accounting Session Id.
- Step 2** If a session exists in the temporary session table for this accounting session id, verification occurs that it is the same subscriber. If it is the same subscriber, an immediate authentication occurs with a response of *Accept* and the remaining steps in this section are omitted. This is considered an Accounting Start Retransmission.
- Step 3** If a session exists in the temporary table for the same accounting session id but for a different subscriber, an immediate response occurs with an accounting response of *Reject* and the remaining steps in this section are omitted.
- Step 4** If a session exists in the database session table for this accounting session id, a verification occurs to determine if it is the same subscriber. If it is the same subscriber, this is treated as a missed stop accounting message and the following occurs:
- A temporary copy of the existing session record is created and the handle of the temporary is passed to the network state machine with an event name of *ntwkRestartSession*. (See section [SP Session Controller Message Processing and State Machine Support](#) for action taken on a *ntwkRestartSession* event.)
  - An immediate authentication response of *ACCEPT* is sent to the 10K BRAS device.
- Step 5** If a session exists in session database table but for a different subscriber, an immediate authentication response of *Reject* is sent and the remaining steps in this section are omitted.
- Step 6** If no session exists in temporary space or in the session database for this subscriber, the following occur:
- Determination is made of the appropriate *temporary* profile to assign, using the NAS IP address and looking up the network profile in the access line entry for that 10K BRAS device. If there is no profile for the 10K BRAS device, an immediate accounting response of *Reject* is sent and the session record is not created.
  - A temporary subscriber network session is created that contains the information in the following table, which was obtained for the profile lookup in step one and the date retrieved from the RADIUS Start.

| Item                                   | Detail                                         |
|----------------------------------------|------------------------------------------------|
| Accounting message:                    |                                                |
| Accounting session id                  |                                                |
| Accounting session handle              |                                                |
| IP address                             |                                                |
| Network device ID                      |                                                |
| Network device interface               |                                                |
| Parameter List                         |                                                |
| Profile Indicator (of default profile) |                                                |
| Userid – username@domain               |                                                |
| Username                               |                                                |
| Domain                                 |                                                |
| Primary Account                        | Set to false no sub-accounting on network side |



| Item              | Detail                               |
|-------------------|--------------------------------------|
| Dependent Session | Set to null for network side session |
| Session State     | Set to <i>Network Created State</i>  |

- Step 7** The handle to the temporary session record is passed to the network state machine with an event name of *ntwkStartSession*. (See section entitled [SP Session Controller Message Processing and State Machine Support](#) for action taken on a *ntwkRestartSession* Event.)
- Step 8** An immediate Accounting Response of Accept to the 10K BRAS device occurs.

## RADIUS Accounting Proxy/Accounting Server Stop Accounting Processing

- Step 1** The *Accounting Session Id* attribute determines if there is a session entry for this subscriber. Both the temporary table and the database are checked for an existing session for this *Accounting Session Id*.
- Step 2** If a session exists in the temporary session table for this accounting session id delete action field in the temporary record is set to true to indicate to the Accounting Start Processing that a Stop Accounting was received while the subscriber was logging in. The system immediately responds with an accounting response of Accept and omits the remaining steps in this section. This is case is considered a Stop Accounting *crossing* a Start Accounting. When the login process has completed, the delete action field in the temporary record is checked. If it is set to *true*, the subscriber is immediately logged out.
- Step 3** If a session entry exists in the policy database for this accounting session id, this is a standard logout and the following action is taken:
- The current session record entry is copied from the policy database into a temporary session record for state machine use in processing.
  - The handle to the temporary session record is passed to the network state machine with an event name of *ntwkEndSession*. (See section 3.4 for action taken on a *ntwkEndSession* event.)
  - An immediate Accounting Response of *Accept* is sent to the 10K BRAS device.
- Step 4** If no session entry exists in the policy database for this accounting session id, this is considered an Accounting Stop Retransmission or a *missed* Accounting Start. An immediate Accounting Response of *Accept* is sent to the 10K BRAS device.

## RADIUS Accounting Proxy/Accounting Server Accounting On/Off Processing

- Step 1** If either an Accounting On or an Accounting Off message is detected by the NP BPM, this indicates that the 10K BRAS device is coming down (Accounting Off) or coming back up (Accounting On).
- Step 2** All of the session entries (active or transient) are deleted.
- Step 3** A broadcast message reporting the failure is sent through the NP Director BPM to the SP associated with the 10K BRAS device.

# Application Network State Machine

This section describes the action taken for each event the network state machine receives. The action taken by the NP Session Controller BPM state machine depends on the state of the subscriber session when the event is received. The network state machine is invoked from three services:

- *genericRadiusStartAccounting*
- *genericRadiusStopAccounting*
- *subscriberServiceToNetworkEvent* (The NP Director BPM uses this service to pass on SP Session Controller BPM events)

## Event - *ntwkStartSession*

The *ntwkStartSession* event is sent from the *genericRadiusStartAccounting* service processing. Only one valid state exists for a *ntwkStartSession* event to be processed in, the *Network Session Created* state, which exists only in the temporary (transient) session record. The following section describes the action taken in this state. It also describes the action taken, if this event is received in any other state. (All other states are deemed invalid.)

### Network Session Created

This section defines the process for providing the 10K BRAS device with the initial *bare bones* profile that is assigned until the service provider BPM indicates the actual default profile this subscriber receives as part of the package he or she purchased.




---

**Note** Any use of temporary records is removed, if an error occurs anywhere in the following process.

---

- Step 1** The network profile id set up during the RADIUS processing locates the list of install and uninstall service rules for that profile and store them in the current profile list field in the temporary session record. If a failure occurs in the process, an error is returned and any remaining steps in this section are omitted.
- Step 2** The information the scheduler requires to program the set of device rules found in [Step 1](#) is created. This includes determining if the rules are programmed through the CLI or through the ISG interface. If the device rules are programmed through the CLI, the scheduler pin is set with the correct information for the scheduler to invoke the *deviceConfig* service. If the device rules are programmed through the ISG (RADIUS VSA) interface, the scheduler pin is set with the correct information for the scheduler to invoke the *deviceConfigISG* service. If a failure occurs during the process, an error is returned and any remaining steps in this section are omitted.
- Step 3** An official session record for this subscriber is created in the Cisco policy database by copying all of the information stored in the temporary session record into the corresponding entries in the session record table entry for the subscriber. The state of the session upon its creation in the policy database is the *Network Install Complete* state.
- Step 4** To program the 10K BRAS device, the scheduler is invoked using the scheduler pin information. The scheduler returns a result of *OK*, or it indicates that it experienced an error trying to program the 10K BRAS device. If an error occurred, the session record is removed from the database and no further processing occurs. If the result returned was *OK*, the process continues to [Step 5](#).
- Step 5** The intercommunication service *subscriberNetworkToServiceRequest* is asynchronously invoked to indicate to the corresponding SP Session Controller BPM that a new subscriber has come onto the network. This is done by sending a *serviceSessionBegin* event through the NP Director BPM to the appropriate SP BPM. (The NP Director BPM uses the network device id to locate the appropriate SP BPM.)
- Step 6** The delete flag is checked to see if an Accounting Stop was received during subscriber login. If the delete action flag is set to false, the process continues to [Step 7](#). If the delete action flag is set to true, the subscriber is cleanly logged out by a call back into the state machine with a *ntwkEndSession* event.

**Step 7** The temporary session record the state machine used for processing is deleted.

### Invalid State

An error is returned to the outside function that invoked the state machine and the temporary session record is removed for that subscriber.

### Event - *ntwkInstallDefaultProfile*

This event is sent from the *subscriberServiceToNetworkEvent* service processing. This event is processed only for a subscriber who has a session record in the policy database. The *subscriberServiceToNetworkEvent* service processing is responsible for putting a copy of the current session record entry, stored in the policy database, into a temporary session record for the state machine to work with.




---

**Note** If an error occurs in the following process, any use of temporary records is removed.

---

There is only one valid state for a *ntwkInstallDefaultProfile* event to be processed in and that is the *Network Install Complete* state. This section also describes the action taken, if this event is received in any other state. (All other states are deemed invalid.)

### Network Install Complete

The NP Session Controller BPM receives the *ntwkInstallDefaultProfile* message, forwarded to it from the SP Session Controller BPM through the NP Director BPM, and performs the following steps.

- 
- Step 1** The network profile id that the SP Session Controller BPM passed in the message locates the list of install and uninstall rules for that profile. The rules are stored in the previous profile list field in the temporary session record. If a failure occurs during the process, an error is returned and the remaining steps in this section are omitted.
  - Step 2** The information is created that the scheduler requires to uninstall the current configuration the 10K BRAS device has. The information is created that the scheduler requires to install the new profile device rules. The scheduler pin information is created by first setting up the uninstall device rules, using the information in the current network profile list in the temporary session record and setting up the install device rules, using the information that was placed in the previous network profile list. The uninstall and install device rules are programmed in one invocation of the scheduler and given to the scheduler as one list. If the device rules are programmed through the CLI, the scheduler pin is set up with the correct information for the scheduler to invoke the *deviceConfig* service. If the device rules are programmed through the ISG (RADIUS VSA) interface, the scheduler pin is set up with the correct information for the scheduler to invoke the appropriate *deviceConfigISG* service.
  - Step 3** The scheduler is invoked using the scheduler pin information to program the 10K BRAS device to uninstall the current profile. The scheduler returns a result of *OK*, or it indicates that it experienced an error trying to program the 10K BRAS device. If an error occurs, the session record is placed in the *Network Out of Sync State*. If the result is *OK*, the session is deleted from the policy database. In either case, success or failure, the process continues to [Step 4](#).
  - Step 4** The temporary session record state is set to *Active* and the temporary session record is copied back to the policy database. The inter communication service *subscriberNetworkToServiceRequest* is asynchronously invoked to indicate to the corresponding SP Session Controller BPM that the subscriber

default profile is in place. This is done by sending a *serviceSessionDefaultProfileInstalled* event through the NP Director BPM to the appropriate SP BPM. (The NP Director BPM uses the network device id to locate the appropriate SP BPM.)

- Step 5** For error processing, the intercommunication service *subscriberNetworkToServiceRequest* is asynchronously invoked to indicate to the corresponding SP Session Controller BPM that the default profile was not successfully applied to the 10K BRAS device. This is done by sending a *serviceSessionDefaultProfileNotInstalled* event through the NP Director BPM to the appropriate SP BPM. (The NP Director BPM uses the network device id to locate the appropriate SP BPM.)

## Invalid State

- 
- Step 1** The intercommunication service *subscriberNetworkToServiceRequest* is asynchronously invoked to indicate to the corresponding SP Session Controller BPM that the default profile was not successfully applied to the 10K BRAS device. This is done by sending a *serviceSessionDefaultProfileNotInstalled* event through the NP Director BPM to the appropriate SP BPM. (The NP Director uses the network device id to locate the appropriate SP BPM.)

## Event - *ntwkEndSession*

The following are the three states that a *ntwkEndSession* event is processed in:

- Network Install Complete
- Active
- Logoff In Progress

These states are tracked in a database session record table. The following section describes the action taken when the *ntwkEndSession* event is received in these states. All other states are deemed invalid. This section also describes the action taken if this event is received in another state.

## Network Install Complete and Active

- 
- Step 1** The device rules set that are needed to properly uninstall the active profile are determined from the session record current network profile list field. The uninstall device rule list creates the appropriate pin information for scheduler use to obtain the current profile uninstalled from the 10K BRAS device.
- Step 2** The session state is set to *Logoff In Progress* in the session record in the policy database. The information is created that the scheduler requires to uninstall the current configuration on the 10K BRAS device. The scheduler pin information is created by locating the uninstall device rules in the session record, contained in the current network profile list field. The uninstall device rules are programmed in one invocation of the scheduler and passed to the schedule in one list. If the device rules are programmed through the CLI, the scheduler pin is given the information for the scheduler to invoke the *deviceConfig* service. If the device rules are programmed through the ISG (RADIUS VSA) interface, the scheduler pin is given the information for the scheduler to invoke the appropriate *deviceConfigISG* service.
- Step 3** To program the 10K BRAS device to uninstall the profile, the scheduler is invoked using the scheduler pin information. The scheduler return a result of *OK* or it indicates that it experienced an error trying to program the 10K BRAS device. If an error occurs, the session record is placed in the *Network Out of Sync State*. If the result is *OK*, the session is deleted from the policy database. In either case, the process continues with [Step 4](#).

- Step 4** To indicate the corresponding SP Session Controller BPM that the session is being logged out, the intercommunication service *subscriberNetworkToServiceRequest* is asynchronously invoked. This is done by sending a *serviceSessionEnd* event through the NP Director BPM to the appropriate SP Session Controller BPM (The NP Director BPM uses the network device id to locate the appropriate SP Session Controller BPM).

## Logoff In Progress

This state indicates the middle of a subscriber logoff; no action is be taken.

## Invalid State

An error is returned but no direct action is taken.

# SP Session Controller Message Processing and State Machine Support

This section describes the action taken for each event the SP Session Controller BPM session state machine receives. The action taken by the SP Session Controller BPM state machine depends on the state of the subscriber session when the event is received.

- 
- Step 1** The network state machine is invoked from the following service:
- *subscriberServiceProcessNetworkEvent*

The NP Director BPM uses this to pass on NP BPM events

## Event - *serviceSessionBegin*

No subscriber session record yet exists, so there is no state check for this event. [Step 1](#) of this section ensures there is no temporary subscriber session record and no subscriber session record in the policy database. The SP Session Controller BPM receives the *serviceSessionBegin* forwarded from the NP Director BPM. Upon receiving a *serviceSessionBegin* event, the SP Session Controller BPM performs the steps below.



### Note

If any of the following steps fail, the SP Session Controller BPM detects a login failure, deletes the memory copy of the session record, and forwards a *ntwkBeginSessionFailed* message to the NP BPM.

- 
- Step 1** A determination is made to determine that the session Id that was passed from the NP Session Controller BPM and has an associated session record it (temporary or in the policy database). If no session record is being tracked for that id, a determination is made to ensure that this userid/ipaddress pair does not have an existing session record (temporary or in the policy database). If no session record is found, the process continues with [Step 2](#).

If a session record is found for this subscriber or session id, this is a service session restart.

- Step 2** The subscriber is authenticated against the policy database subscriber table.

- Step 3** The default network and service profiles for the subscriber are located. The SP Session Controller BPM can access all information necessary for the subscriber to receive the service and network treatment defined by his service level (package). The Cisco application uses a hierarchical method to locate a subscribers network and service profiles to guarantee that the subscriber receives the proper “base treatment” at login. Finding a subscriber base/login treatment is determined from the subscriber table, if there is a defined network or service profile. If profiles exist, they are used. Otherwise, the subscriber group is checked to determine if a network or service profile is defined for the group. If one is defined, it is used. Otherwise, the subscriber domain (if there is a domain) is checked to determine if there are any defined profiles for the domain. If defined profiles exist, they are used. If not, the subscriber is not allowed to log into the retail side of the configuration. (Only a service profile is required for a subscriber if no network profile is found. This indicates that the default profile, applied on the wholesale side, is used as the default)
- Step 4** The service profile is checked to determine if it is marked as a service bundle. If it is, the process continues with [Step 5](#). Otherwise, the process continues at [Step 7](#).
- Step 5** The quota and DPI profile information in the subscriber service profile are used to correctly arrange the DPI device information in the temporary session record. The subscriber current usage is compared to the quota information stored in the service bundle to determine if the subscriber is in breach. If the subscriber is in breach, the process logs into the DPI device using the breach profile indicated in the service profile (bundle). If there is a new network profile defined for the breach state update, the profile id in the temporary record is used as the *default* profile id. If the subscriber is not in breach, the process logs in to the DPI device using the default DPI Profile indicated in the service profile (bundle).
- Step 6** The subscriber usage table is furnished with the appropriate quota information.
- Step 7** Service event information found in the subscriber service profile/bundle ([Step 3](#)) obtains all services that this subscriber can access to during his login. The information is stored in the appropriate places in the temporary session record. This step is performed whether or not there is any DPI processing necessary. The three categories of service events listed below are supported for this implementation and have their own place in the temporary (and eventually the policy database entry) session record:
- On-Demand Services (service subscriber can access)
  - Absolute Time-based Services (services scheduled to run at a particular time)
  - Relative Time-based Services (services scheduled to run at a certain time relative to subscriber login).
- Step 8** The inter communication service *subscriberNetworkToServiceRequest* is asynchronously invoked with a *ntwkInstallDefaultProfile* event and with the *profileId* for the appropriate network profile that is applied as the subscriber default profile. (The network profile was determined in [Step 3](#).)

**Note**


---

If no network profile is found at any level of the subscriber hierarchy, this message is sent with a *profileId* value of zero.

---

**Note**


---

The session record remains in temporary space until the NP indicates it is active, because the DPI device is not yet affected. All pertinent information has been stored, but has not gone to the DPI Device.

---

## Event - *serviceSessionDefaultProfileInstalled*

SP Session Controller BPM receives the *serviceSessionDefaultProfileInstalled* event forwarded from the NP Session Controller BPM through the NP Director BPM. The SP Session Controller BPM is responsible for the following:

- 
- Step 1** Locates the temporary service login session record associated with the session. A check determines if this is a DPI case (service profile or bundle). If it is a service bundle, the process continues at [Step 2](#). If it is not a bundle, there is no DPI treatment needed. The process continues at [Step 3](#).
  - Step 2** From the data stored when the *serviceSessionBegin* we processed, determines the DPI vendor logged into with the appropriate DPI Profile.
  - Step 3** Determines the scheduler pin information for any time-based services that this subscriber inherits from the service profile/bundle he is associated with. This includes setting up the scheduler information for any relative time-based jobs and any absolute time-based jobs.
  - Step 4** Creates the service provider session record in the policy database by copying it from the temporary service login session into a Cisco policy DB session record. When the session record is copied into the policy database, sets the session state to *Active*, indicating that external applications can access these sessions.

### Event - *serviceSessionEnd*

- 
- Step 1** The subscriber service profile is located. A check determines if the profile has the bundle field selected, indicating that DPI involvement is required. A check determines if the subscriber profile is a bundle. If the subscriber profile is a bundle, the process continues to [Step 2](#) to ensure the subscriber is properly *logged* out of the DPI device. Otherwise, the process continues at [Step 3](#).
  - Step 2** The device information is located for the DPI that the subscriber is currently logged into. Using the vendor information to identify the DPI device type, appropriate action is taken to log the subscriber out of the DPI device. If a failure occurs during the logout process, the session state is set to *Service Out of Sync*, an error is returned, and the process does not proceed to [Step 3](#).
  - Step 3** The session is deleted from the policy database. If a failure occurs during this step, the session state is set to *Service Out of Sync*, an error is returned, and the process does not proceed to [Step 4](#).
  - Step 4** An end session scheduler pin configuration is created for use in programming the scheduler to unschedule any of the time-based jobs (absolute or relative) for the subscriber.
  - Step 5** The end session operator of the scheduler using the pin information created in [Step 4](#) is invoked. If this operation fails, no action is taken.



---

**Note** The session record is left in the *Service Out of Sync*.

---

## Inter BPM Communication

A SOAP interface is used for all inter-BPM communication. The methods described indicate the format for the events passed between the NP Session Controller BPM and the SP Session Controller BPM during subscriber login and logoff.

## NP Session Controller BPM

### Service - *subscriberNetworkToServiceRequest*

The *subscriberNetworkToServiceRequest* service invokes a SOAP listener service on the NP Director BPM.

- Inputs: directorServiceToInvoke
  - directorsoaphostport (hostname:soap port configured in BPM Configuration)
  - eventname
  - ipaddress
  - networkdeviceid
  - sessionid
  - soapservicename (service name specified in the soap agent defined as the listener)
  - userid
- Outputs: result (OK or error message)
- WSDL file needed: networkDirectorListener.wsdl
- Listener Service on NP Director: *subscriberNetworkToServiceEvent*

### Service - *subscriberServiceToNetworkEvent*

The *subscriberServiceToNetworkEvent* -event is a SOAP listener service that is invoked from the NP Director BPM to deliver events from the SP Session Controller.

- Inputs:
  - eventname
  - networkdeviceid
  - networkprofileid
  - sessionid
- Outputs: result (OK or error message)
- WSDL file generated: networkBPMServiceEventListener.wsdl
- SP Director BPM uses to invoke this service: *deliverSubscriberServiceEvent*

### Subscriber-based Events from the NP Session Controller BPM to the SP BPM

All subscriber-based messages coming from the NP through the director to the SP Session Controller contain, minimally, the NAS IP-Address the subscriber came in on. The 10K BRAS device address is used by the NP Director BPM to find the SP Session Controller BPM that the subscriber is associated with.

- *serviceSessionBegin*
- *serviceSessionDefaultProfileInstalled*
- *serviceSessionDefaultProfileNotInstalled*
- *serviceSessionEnd*



## SP Session Controller BPM

### Service - *subscriberServiceToNetworkRequest*

The *subscriberServiceToNetworkRequest* invokes a SOAP listener service on the NP Director BPM.

- Inputs: *directorServiceToInvoke*
  - *directorsoaphostport* (hostname:soap port configured in BPM Configuration)
  - *eventname*
  - *networkdeviceid*
  - *networkprofileid*
  - *sessionid*
  - *soapservicename* (service name specified in the soap agent defined as the listener)
- Outputs: result (OK or error message)
- WSDL file needed: *networkDirectorServiceListener.wsdl*
- Listener Service on NP Director: *subscriberServiceToNetworkEvent*

### Service - *subscriberServiceProcessNetworkEvent*

The *subscriberServiceProcessNetworkEvent* service is a the SOAP listener service that is invoked from the NP Director BPM to deliver events from the SP Session Controller BPM.

- Inputs:
  - *eventname*
  - *networkdeviceid*
  - *ipaddress*
  - *sessionid*
  - *userid*
- Outputs: result (OK or error message)
- WSDL file generated: *serviceBPMNetworkEventListener.wsdl*
- Service NP Director BPM uses to invoke this service: *deliverSubscriberNetworkEvent*

### Subscriber-based events from the SP Session Controller BPM to the NP Session Controller BPM

- *ntwkInstallDefaultProfile*
- *ntwkBeginSessionFailed*

## NP Director BPM

### Service - *subscriberNetworkToServiceEvent*

The *subscriberNetworkToServiceEvent* service is the SOAP listener service on the NP Director BPM that accepts events from the NP BPM.

- Inputs: eventname
  - networkdeviceid
  - ipaddress
  - sessionid
  - userid
- Outputs: result (OK or error message)
- WSDL file generated: *networkDirectorListener.wsdl*

### Service - *subscriberServiceToNetworkEvent*

The *subscriberServiceToNetworkEvent* service is the SOAP listener service on the NP Director BPM that accepts events from the SP Session Controller BPM to deliver events to the SP Session Controller BPM.

- Inputs: eventname
  - networkdeviceid
  - networkprofileid
  - sessionid
- Outputs: result (OK or error message)
- WSDL file generated: *networkDirectorServiceListener.wsdl*

### Service - *deliverSubscriberNetworkEvent*

- Inputs: *servicetoinvoke*
  - soaphostport (hostname:soap port configured in BPM Configuration)
  - eventname
  - ipaddress
  - networkdeviceid
  - sessionid
  - soapservicename (service name specified in the soap agent defined as the listener)
  - userid
- Outputs: result (OK or error message)
- WSDL file used: *serviceBPMNetworkEventListener.wsdl*

### Service - *deliverSubscriberServiceEvent*

*deliverSubscriberServiceEvent-*

- Inputs: *servicetoinvoke*
  - soaphostport (hostname:soap port configured in BPM Configuration)
  - eventname
  - networkdeviceid

- sessionid
- soapservicename (service name specified in the soap agent defined as the listener)
- networkprofileid
- Outputs: result (OK or error message)
- WSDL file generated: *networkBPMServiceEventListener.wsdl*

There are currently no events that are initiated by the NP Director BPM. All events dealt with by the NP Director BPM deals are events it must to forward from either the NP Session Controller BPM or the SP BPM.



# External Application Interface Communications

---

## Overview

This section discusses the communications between external applications and the SP Director BPM that affect subscriber sessions on the SP Session Controller BPM. The NP Session Controller BPM initiates and takes down a session with the SP through a SOAP interface. The subscriber is initially identified across the wholesale-retail boundary via his username, domain, and IP address, and, in some cases by his session id. Cisco does not support domainless access. A subscriber must belong to a domain.

If the service profiles have DPI profiles, the following conditions apply:

- The P-Cube interface does not support tiers.
- The P-Cube interface has only one active profile at a time for each subscriber.
- When a DPI profile is involved, temporary and permanent profiles cannot be mixed when applying tiers for a subscriber.
- Service profiles containing DPI profiles must be removed in the reverse order they were applied.
- Service profiles cannot have multiple service events with DPI profiles.

However, if the service profiles do not have DPI profiles, none of these conditions apply. No limits exist on network profile tiering.

## Session Controller Registration with SP Director BPM

When the SP Session Controller BPM receives a *serviceSessionBegin* message from the NP Session Controller BPM, it determines if the subscriber is already registered with the SP Director BPM. The SP Session Controller BPM determines if an external id exists in the subscriber policy database entry. If this id does not exist, the SP Session Controller BPM allocates an external device identifier and communicates the appropriate information to the SP Director BPM via the Policy Repository BPM.

## External Application Registration

This section describes the interface communication and the API used for that interaction. The SP Director API being exposed is the direct interface in which external applications can affect subscribers and services contained on the Cisco SP BPMs. The API contains the available operations.

All interactions between the external application and the SP Director BPM occur via SOAP messages. These are asynchronous messages returned while performing certain actions. This is due to the nature of some executed services that may require NP BPM interactions. Here, an immediate synchronous

response may not be possible on the action successfully completed. An Event Interface between the SP Director BPM and the external applications alerts the external application(s) of a failure that occurred during subsequent processing of the external application request.

Although fields may be defined as *mandatory* or *optional*, ALL fields must pass through each API call. The optional fields do not NEED a value. They can be set to the empty string, or to "".

### Service - *registerApplication*

The external application registers itself with the SP Director BPM to receive potential error messages, as described earlier. All API requests are verified against the external application registration; therefore, application registration is mandatory. Table [Table 1-1](#) describes the functions and parameters for the *registerApplication* service.

**Table 1-1 Functions and Parameters - registerApplication**

| Parameter/<br>Return | Name          | Mandatory/<br>Optional | Function                                                                                                                                      |
|----------------------|---------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter            | appid         | Mandatory              | The unique id that identifies this external application in the BPM system.                                                                    |
| Parameter            | endpoint      | Mandatory              | The URL formatted location to which SOAP messages are sent for notifications to the external application.                                     |
| Parameter            | eventtype     | Optional               | The event type to notify the external application about. If this is set to null, currently defaults to <i>subscriberProfileFailureEvent</i> . |
| Parameter            | eventcriteria | Optional               | The criteria used to specify what events are sent back within an event type. The default is <i>ANY</i> .                                      |
| Return               | result        |                        | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message.                                   |

### Service - *unregisterApplication*

The unregister function allows the external application to remove itself (or a registered event type) from the SP Director BPM. [Table 1-2](#) describes *unregisterApplication* functions and parameters.

**Table 1-2 Functions and Parameters - unregisterApplication**

| Parameter/<br>Return | Name      | Mandatory/<br>Optional | Function                                                                                                                                      |
|----------------------|-----------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter            | appid     | Mandatory              | The unique id that identifies this external application in the BPM system.                                                                    |
| Parameter            | eventtype | Optional               | The event type to notify the external application about. If this is set to null, currently defaults to <i>subscriberProfileFailureEvent</i> . |
| Return               | result    |                        | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message.                                   |

## Utility Functions

API calls made via the external application that modifies user configuration use a unique subscriber index or id that represents the subscriber on the BPM devices. The id is generated via the Cisco devices per username, domain, and IP address. Table 1-3 describes *subscriberIndexLookupForUser* functions and parameters required to find the subscriber id.

### Function - *getSubscriberExternalIndex*

Table 1-3 Functions and Parameters - *getSubscriberExternalIndex*

| Parameter/<br>Return | Name         | Mandatory/<br>Optional | Function                                                                                                    |
|----------------------|--------------|------------------------|-------------------------------------------------------------------------------------------------------------|
| Parameter            | appid        | Mandatory              | The unique identifier that indicates this external application in the BPM system.                           |
| Parameter            | username     | Mandatory              | The subscriber login name.                                                                                  |
| Parameter            | domain       | Mandatory              | The domain the subscriber belongs to.                                                                       |
| Parameter            | ipaddress    | Mandatory              | The IP address assigned to this subscriber.                                                                 |
| Return               | result       |                        | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message. |
| Return               | subscriberid |                        | The unique user identifier the action is performed on.                                                      |

## Profile Processing Service

Two functions allow changes to a current subscriber configuration:

- *applyProfile*
- *releaseProfile*

The *applyProfile* service affects only the profiles (network 10K BRAS device-based or DPI-based); The *releaseprofile* may affect a profile while invoking another service as well. The profile modifications are used to change or modify subscriber profiles only.

### Function - *applyProfile*

The *applyProfile* functions are used to make a change, temporary or permanent, to an existing base level network (10K BRAS device) or DPI profile. A temporary change takes effect for an active session and is released in one of the following ways:

- A duration provided through the API expires.
- The external application that applied the profile manually releases it, using the *releaseProfile* API.

A permanent profile change takes effect with, or without, an active subscriber session. The profile is added to the subscriber profile list. It takes effect upon every login, until it is released. A permanent profile change is released by one of the following:

- Manual release by the external application that applied the profile using the *releaseProfile* API.
- The subscriber logs out. (It is reapplied when the subscriber logs in again.)

The network and DPI profiles applied using this interface are defined in a subscriber access service that is attached to a subscriber access service event in a subscriber access service profile. The service event is defined to run immediately and must have at least a network or DPI profile defined in the service that is used to create the service event. If the service contains only a network profile, only the network-side BRAS is affected. If the service contains only a DPI profile, only the service-side DPI is affected. If both a network profile and a DPI profile are configured in the service that is used to create the service event, an action is taken on both the network-side BRAS and the service-side DPI. If there is neither a network or DPI profile in the service, no action is taken.

Table 1-4 describes the *applyProfile* functions and parameters. These are used when changing or modifying subscriber profiles only.

**Table 1-4 Functions and Parameters - applyProfile**

| Parameter/<br>Return | Name           | Mandatory/<br>Optional | Function                                                                                                                                                                                                                                                                                       |
|----------------------|----------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter            | appid          | Mandatory              | The unique identifier that indicates this external application in the BPM system.                                                                                                                                                                                                              |
| Parameter            | subscriberid   | Mandatory              | The unique identifier of the user that the action is performed on. This is obtained by the previously defined <i>subscriberIndexLookupForUser</i> function.                                                                                                                                    |
| Parameter            | permanent      | Mandatory              | This indicates if this is a permanent change to the subscriber, indicating that the profile should be used on next login, not just for this session. Valid values are <i>true</i> or <i>false</i> .                                                                                            |
| Parameter            | tiered         | Mandatory              | This indicates if the profile being applied is to be a tier, or built on a base amount of existing configuration rather than replacing the old profile with a new one. Valid values are <i>true</i> or <i>false</i> .                                                                          |
| Parameter            | serviceprofile | Mandatory              | This is the name of the service profile. The service profile contains an immediate service event, where the service used contains a network profile, a DPI profile, or both profiles.                                                                                                          |
| Parameter            | duration       | Optional               | The amount of time, in seconds, for which this profile is applied. A value of 0 indicates that a manual release or session logout are the only ways to release the profile. A non-zero value represents the time, in seconds, that the profile is applied before it is automatically released. |
| Return               | result         |                        | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message.                                                                                                                                                                                    |
| Return               | transactionid  |                        | The unique identifier returned that identifies the action recently performed. Its potential use is for <i>releaseProfile</i> or <i>alterProfile</i> , if desired                                                                                                                               |

### Function - *releaseProfile*

The *releaseProfile* function releases a profile that was placed by the *applyProfile* API, described earlier. The *releaseProfile* application ID (*appid*) must match the application id that applied the profile. If it does not, an error occurs. Table 1-5 describes the *releaseProfile* functions and parameters.



Table 1-5 Functions and Parameters - releaseProfile

| Parameter/<br>Return | Name           | Mandatory/<br>Optional | Function                                                                                                                                                                       |
|----------------------|----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter            | appid          | Mandatory              | The unique id that identifies this external application in the BPM system.                                                                                                     |
| Parameter            | subscriberid   | Mandatory              | The unique identifier of the user that the action is performed on. This is obtained by the <i>subscriberIndexLookupForUser</i> function.                                       |
| Parameter            | transactionid  | Mandatory              | The id returned during the initial <i>applyProfile</i> call. This id is passed back when the <i>alterProfile</i> or <i>applyProfile API</i> was invoked to place the profiles. |
| Parameter            | serviceprofile | Mandatory              | The name of the service profile the user wants to release.                                                                                                                     |
| Return               | result         |                        | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message.                                                                    |

## Feature Processing Service

The service in this section invoke specific services created on the Cisco BPM devices. This service performs a variety of tasks and can change and modify potential profiles, both network and DPI.

### API - *applyProfileFeature*

The *applyProfileFeature* API invokes a service on the SP Session Controller BPM. The *applyProfileFeature* API may invoke the following services:

- *changeSubscriberServiceLevel*
- *changeSubscriberServiceGroup*
- *supplementalQuota*

Table 1-6 describes *applyProfileFeature* parameters and functions.

Table 1-6 Functions and Parameters - applyProfileFeature

| Parameter/<br>Return | Name         | Mandatory<br>/Optional | Function                                                                                                                                                                                                                                                                                   |
|----------------------|--------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter            | appid        | Mandatory              | The unique id that identifies this external application in the BPM system.                                                                                                                                                                                                                 |
| Parameter            | subscriberid | Mandatory              | The unique identifier of the user that the action is performed on. This is obtained by the previously defined <i>subscriberIndexLookupForUser</i> function.                                                                                                                                |
| Parameter            | featurename  | Mandatory              | This indicates the name of the service defined on the BPM device to be invoked. Only the following are allowed: <ul style="list-style-type: none"> <li>• <i>changeSubscriberServiceLevel</i></li> <li>• <i>changeSubscriberServiceGroup</i></li> <li>• <i>supplementalQuota</i></li> </ul> |

|           |               |          |                                                                                                                                                                                                                                                                                          |
|-----------|---------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter | duration      | Optional | The amount of time in seconds for which the service shall run. A value of 0 indicates that a manual release or session logout is the only way to release the profile. A non-zero value represents the time, in seconds, that the profile is applied before it is automatically released. |
| Parameter | duration      | Optional | The amount of time in seconds for which the service shall run. A value of 0 indicates that a manual release or session logout is the only way to release the profile. A non-zero value represents the time, in seconds, that the profile is applied before it is automatically released. |
| Parameter | dataValue1    | Optional | This parameter is used differently for different features and services that are being invoked through this API. Refer to the next section for the specific feature or service that uses this field.                                                                                      |
| Parameter | dataValue2    | Optional | This parameter is used differently for different features and services that are being invoked through this API. Refer to the next section for the specific feature or service that uses this field.                                                                                      |
| Parameter | dataValue3    | Optional | This parameter is used differently for different features and services that are being invoked through this API. Refer to the next section for the specific feature or service that uses this field.                                                                                      |
| Parameter | dataValue4    | Optional | This parameter is used differently for different features and services that are being invoked through this API. Refer to the next section for the specific feature or service that uses this field.                                                                                      |
| Return    | result        |          | This returns <i>OK</i> , if the function was executed successfully; otherwise, it returns an error message.                                                                                                                                                                              |
| Return    | transactionid |          | The unique identifier returned that identifies the action recently performed. It is used to potentially <i>releaseProfile</i> or <i>alterProfile</i> , if desired.                                                                                                                       |

## Supported Features and Services Data Values

### Service - *changeSubscriberServiceLevel*

Table 1-7 *changeSubscriberServiceLevel* Parameters

| Parameter  | Description                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------|
| dataValue1 | The group name the subscriber is to inherit its service profile and network profile characteristics from. |
| dataValue2 | "" Not used; set to null string.                                                                          |
| dataValue3 | "" Not used; set to null string.                                                                          |
| dataValue4 | "" Not used; set to null string.                                                                          |

**Service - *changeSubscriberServiceGroup*****Table 1-8** *changeSubscriberServiceGroup* Parameters

| Parameter  | Description                                                 |
|------------|-------------------------------------------------------------|
| dataValue1 | The group name the subscriber is to be physically moved to. |
| dataValue2 | "" Not used; set to null string.                            |
| dataValue3 | "" Not used; set to null string.                            |
| dataValue4 | "" Not used; set to null string.                            |

**Service - *supplementQuota***

Only a total volume supplement is allowed, so the only data value this feature requires is the supplement value. The supplement value must match one of the supplement values that was configured in the bundle this subscriber was associated with during login.

**Table 1-9** *supplementQuota* Parameters

| Parameter  | Description                                                                    |
|------------|--------------------------------------------------------------------------------|
| dataValue1 | The supplement value. This is the amount to increment the total allowed quota. |
| dataValue2 | "" Not used; set to null string.                                               |
| dataValue3 | "" Not used; set to null string.                                               |
| dataValue4 | "" Not used; set to null string.                                               |

## Inter BPM Communication

All inter-BPM communication is via a SOAP interface. The methods described in this section indicate the format for the events that are passed between the NP Session Controller BPM and the SP Session Controller BPM, as well as the SP Session Controller BPM and SP Director BPM during Profile/Feature requests from the external application(s).

### SP Session Controller BPM

**Service - *subscriberProcessProfileEvent***

The *subscriberProcessProfileEvent* is a SOAP service listener invoked by a service on SP Director BPM.

Table 1-10 subscriberProcessProfileEvent Parameter Details

| Parameter                           | Type                | Description                                                                                       |
|-------------------------------------|---------------------|---------------------------------------------------------------------------------------------------|
| appid                               | Input               |                                                                                                   |
| datavalue1                          | Input               |                                                                                                   |
| datavalue2                          | Input               |                                                                                                   |
| datavalue3                          | Input               |                                                                                                   |
| datavalue4                          | Input               |                                                                                                   |
| dpiprofile                          | Input               |                                                                                                   |
| duration                            | Input               |                                                                                                   |
| eventname                           | Input               |                                                                                                   |
| featurename                         | Input               |                                                                                                   |
| ntwkprofile                         | Input               |                                                                                                   |
| permanent                           | Input               |                                                                                                   |
| subscriberindex                     | Input               |                                                                                                   |
| tiered                              | Input               |                                                                                                   |
| transactionid                       | Input               |                                                                                                   |
| result                              | Output              | The result is either OK, or an error message is sent.                                             |
| transaction id                      | Output              | This value is returned to the external application.                                               |
| subscriberProfileEventListener.wsdl | Generated WSDL File | *The SP Director BPM uses this file to invoke the <i>subscriberProcessProfileRequest</i> service. |

### Service - *sendPCSRegistration*

The *sendPCSRegistration* is used to invoke a SOAP listener service on the SP Policy Repository BPM.

Table 1-11 sendPCSRegistration Parameter Details

| Parameter          | Type  | Detail                                                                |
|--------------------|-------|-----------------------------------------------------------------------|
| repositorysoapport | Input | The hostname:soap port configured in the BPM configuration.           |
| soapservicename    | Input | The service name specified in the SOAP agent defined as the listener. |
| pcsDBList          | Input |                                                                       |
| pcsHost            | Input |                                                                       |
| pcsListenerPort    | Input |                                                                       |
| pcsPort            | Input |                                                                       |
| pcsRMIPort         | Input |                                                                       |
| pcsType            | Input |                                                                       |

| Parameter                              | Type                                   | Detail                            |
|----------------------------------------|----------------------------------------|-----------------------------------|
| registrationStage                      | Input                                  |                                   |
| repositoryDBPort                       | Output                                 |                                   |
| repositoryRMI                          | Output                                 |                                   |
| result                                 | Output                                 | This is OK, or a message is sent. |
| repositoryPCSServiceEventListener.wsdl | WSDL File Required                     |                                   |
| processPCSRegistration                 | Listener Service on Service Repository |                                   |

### Service - *sendSubscriberIPAddress*

The *sendSubscriberIPAddress* service is used to invoke a SOAP listener service on the SP Policy Repository BPM.

**Table 1-12** *sendSubscriberIPAddress* Detail

| Parameter                              | Type                                     | Detail                                                                |
|----------------------------------------|------------------------------------------|-----------------------------------------------------------------------|
| repositoryservicetoinvoke              | Input                                    |                                                                       |
| repositorysoaphostport                 | Input                                    | The hostname:soap port configured in the BPM configuration.           |
| soapservicename                        | Input                                    | The service name specified in the SOAP agent defined as the listener. |
| externalid                             | Input                                    |                                                                       |
| ipaddress                              | Input                                    |                                                                       |
| result                                 | Output                                   |                                                                       |
| repositoryPCSServiceEventListener.wsdl | Required WSDL File                       |                                                                       |
| updateSubscriberIP Address             | Listener Service on SP Policy Repository |                                                                       |

### Service - *sendCacheRequest*

The *sendCacheRequest* service is used to invoke a SOAP listener service on the SP Policy Repository BPM.

Table 1-13 *sendCacheRequest* Details

| Parameter                              | Type                                         | Detail                                                                |
|----------------------------------------|----------------------------------------------|-----------------------------------------------------------------------|
| repositoryservicetoinvoke              | Input                                        | The hostname:soap port configured in the BPM configuration.           |
| repositorysoapservicename              | Input                                        | The service name specified in the SOAP agent defined as the listener. |
| data                                   | Input                                        |                                                                       |
| databaseType                           | Input                                        |                                                                       |
| reqHost                                | Input                                        |                                                                       |
| reqPort                                | Input                                        |                                                                       |
| result                                 | Output                                       |                                                                       |
| repositoryPCSServiceEventListener.wsdl | Required WSDL File                           |                                                                       |
| processCacheRequest                    | Listener Service on SP Policy Repository BPM |                                                                       |

## SP Policy Repository BPM Interface

### Service - *processPCSRegistration*

The *processPCSRegistration* is a SOAP service listener invoked by a service on the SP Session Controller BPM.

Table 1-14 *processPCSRegistration* Detail

| Parameter                              | Type                | Detail                                                        |
|----------------------------------------|---------------------|---------------------------------------------------------------|
| pcsDBList                              | Input               |                                                               |
| pcsHost                                | Input               |                                                               |
| pcsListenerPort                        | Input               |                                                               |
| pcsPort                                | Input               |                                                               |
| pcsRMIPort                             | Input               |                                                               |
| pcsType                                | Input               |                                                               |
| faultString                            | Output              |                                                               |
| repositoryDBPort                       | Output              |                                                               |
| repositoryRMI                          | Output              |                                                               |
| repositoryPCSServiceEventListener.wsdl | Generated WSDL File |                                                               |
| sendPCSRegistration                    | Service             | Used by the SP Session Controller BPM to invoke this service. |

**Service - *updateSubscriberIPAddress***

The *updateSubscriberIPAddress* service is a SOAP service listener invoked by a service on the SP Session Controller BPM.

**Table 1-15** *updateSubscriberIPAddress* Detail

| Parameter                              | Type                | Detail                                                        |
|----------------------------------------|---------------------|---------------------------------------------------------------|
| externalid                             | Input               |                                                               |
| ipaddress                              | Input               |                                                               |
| faultString                            | Output              |                                                               |
| repositoryPCSServiceEventListener.wsdl | Generated WSDL File |                                                               |
| sendSubscriberIPAddress                | Service             | Used by the SP Session Controller BPM to invoke this service. |

**Service - *processCacheRequest***

The *processCacheRequest* is a SOAP service listener invoked by a service on the SP Session Controller BPM.

**Table 1-16** *processCacheRequest* Detail

| Parameter                              | Type                | Detail                                                    |
|----------------------------------------|---------------------|-----------------------------------------------------------|
| data                                   | Input               |                                                           |
| databaseType                           | Input               |                                                           |
| reqHost                                | Input               |                                                           |
| reqPort                                | Input               |                                                           |
| faultString                            | Output              |                                                           |
| repositoryPCSServiceEventListener.wsdl | Generated WSDL File |                                                           |
| sendCacheRequest                       | Service             | Used by SP Session Controller BPM to invoke this service. |

**Service Director BPM**

This section describes the supported external application interfaces.

**Service - *applyProfile***

The *applyProfile* service is a a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

Table 1-17 *applyProfile* Details

| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| appid                         | Input               |                                         |
| duration                      | Input               |                                         |
| serviceprofile                | Input               |                                         |
| permanent                     | Input               |                                         |
| subscriberid                  | Input               |                                         |
| tiered                        | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| transactionid                 | Output              |                                         |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

**Service - *releaseProfile***

The *releaseProfile* service is a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

Table 1-18 *releaseProfile* Details

| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| appid                         | Input               |                                         |
| subscriberid                  | Input               |                                         |
| transactionid                 | Input               |                                         |
| serviceprofile                | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

**Service - *applyProfileFeature***

The *applyProfileFeature* service is a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

Table 1-19 *applyProfileFeature* Details

| Parameter  | Type  | Detail |
|------------|-------|--------|
| appid      | Input |        |
| datavalue1 | Input |        |
| datavalue2 | Input |        |
| datavalue3 | Input |        |
| datavalue4 | Input |        |
| duration   | Input |        |



| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| featurename                   | Input               |                                         |
| subscriberindex               | Input               |                                         |
| tiered                        | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| transactionid                 | Output              |                                         |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

### Service - *getSubscriberExternalIndex*

The *getSubscriberExternalIndex* service is a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

**Table 1-20** *getSubscriberExternalIndex* Details

| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| appid                         | Input               |                                         |
| domain                        | Input               |                                         |
| ipaddress                     | Input               |                                         |
| username                      | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| subscriberid                  | Output              |                                         |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

### Service - *registerApplication*

The *registerApplication* service is a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

**Table 1-21** *registerApplication* Details

| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| appid                         | Input               |                                         |
| endpoint                      | Input               |                                         |
| eventcriteria                 | Input               |                                         |
| eventtype                     | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

**Service - *unregisterApplication***

The *unregisterApplication* service is a SOAP service listener invoked by the external application(s). If the application id provided is not configured on the SP Director BPM the API request fails.

**Table 1-22** *unregisterApplication* Details

| Parameter                     | Type                | Detail                                  |
|-------------------------------|---------------------|-----------------------------------------|
| appid                         | Input               |                                         |
| eventype                      | Input               |                                         |
| result                        | Output              | Result is OK, or error message is sent. |
| applicationEventListener.wsdl | Generated WSDL File |                                         |

**SP Session Control Interface****Service - *subscriberProcessProfileEvent***

The *subscriberProcessProfileEvent* service is a SOAP service listener invoked by a service on the SP Director BPM.

**Table 1-23** *subscriberProcessProfileEvent* Details

| Parameter                           | Type                | Detail                                            |
|-------------------------------------|---------------------|---------------------------------------------------|
| appid                               | Input               |                                                   |
| datavalue1                          | Input               |                                                   |
| datavalue2                          | Input               |                                                   |
| datavalue3                          | Input               |                                                   |
| datavalue4                          | Input               |                                                   |
| serviceprofile                      | Input               |                                                   |
| duration                            | Input               |                                                   |
| eventname                           | Input               |                                                   |
| featurename                         | Input               |                                                   |
| permanent                           | Input               |                                                   |
| subscriberindex                     | Input               |                                                   |
| tiered                              | Input               |                                                   |
| transactionid                       | Input               |                                                   |
| result                              | Output              | Result is OK, or error message is sent.           |
| transactionid                       | Output              | This is returned to the external application.     |
| subscriberProcessEventListener.wsdl | Generated WSDL File |                                                   |
| subscriberProcessProfileRequest     | Service             | SP Director BPM uses this to invoke this service. |



# Agents

---

## Overview

This chapter discusses the Broadband IP Services Module (BSM) application agents:

- Event
- [PCube](#)
- [RADIUS](#)
- Scheduler
- Socket
- [SOAP](#)
- TCL

## Application Agents

### Event

The Event agent delivers events matching the *ReceiveEvent* operator criteria. If the operator cannot process the event, the event is queued inside the operator. When the *ReceiveEvent* operator is invoked from a service, it checks its internal queue or waits for an event to arrive. The *ReceiveEvent* operator accepts an event only if it matches the filter in its configuration. When a match is found, the operator sends a message with fields matching the properties in the received event, in addition to the event name and class.

When configuring the *ReceiveEvent* operator, the filter must be validated to ensure proper formatting. The Event agent can attempt to contact the event dispatcher to register itself as a listener for the event class. If the connection or registration is unsuccessful, a *RegistrationFailedException* is thrown. Multiple *ReceiveEvent* operator instances can listen for the same event. Every operator instance receives notification of the event when it arrives.

## Netconf

The Netconf agent interacts with network devices, such as routers and firewalls. It parses the configuration scripts into device configuration commands, sends commands to the devices, and receives responses from the devices. The Netconf agent has two operators, `netconf` and `config`. The `netconf` operator has the parser. The `netconf` operator has three input pins: `script`, `device access info`, and `input parameters`. The `netconf` operator has two output pins: `result` and `output parameters`.

## PCube

The functionality of the *pcubeAgent* includes operators, usage properties, statistics, and packaging. A P-Cube device can perform deep-packet inspection and manipulation. Primarily, the BPM utilizes the following functionality:

Quota restriction

Traffic trend analysis

The P-Cube system comprises two pieces. The Service Engine (SE) is a 2U appliance that runs the applications (packages in P-Cube parlance) and inspects and manipulates the data that passes through it. The Subscriber Manager (SM) is a software application that runs on an ancillary SPARC machine that manages subscribers on the SE. The SE communicates with other system through a stateless, proprietary TCP protocol. The SE sends Raw Data Records (RDRs) over this protocol to a device, specified to the SE during administration. The communication to and from the BPM is through the *pcubeAgent*. The Engage application of a P-Cube appliance (SE) tracks user traffic utilization information, based on the IP address for each logged on subscriber.

## RADIUS

The Remote Authentication Dial In User Service (RADIUS) protocol is a security service for authenticating and authorizing dial-up users. In the BSM offering, the BPM device supports a RADIUS server function. RADIUS traffic is sent directly from the 10K BRAS device to the BPM-Session-controller. This includes start, stop, and interim messages, which require acknowledgements to be used.

For the BSM offering, sessions are tracked via RADIUS. Wholesale RADIUS accounting start packets include a mapping of IP address to LineID. The SP has no other mechanism to learn the subscriber presence or removal from the network, except for the NP Session Controller BPM, which indicates that a RADIUS login occurred for a subscriber.

## Scheduler

The Scheduler allows the BPM system to schedule the delayed triggering of services and policies. The Scheduler supports functionality for specifying absolute or relative start time, recurrence, and duration. Each job comprises a set of *do* and *undo* commands. Each command has a step value that allows the Scheduler to perform them in the desired order. If one command in a job fails, the Scheduler enacts the *undo* commands (if provided) that have a step value less than, or equal to, the step value of the offending command. These *undo* commands are enacted in descending step order as opposed to the increasing order of the *do* commands.

## Socket

The Socket agent maintains as little state about the clients as possible. An internal division of the agent separates the agent into an engine-side and a client-side. The division allows the engine side to run at the engine speed and the client side to run faster, to handle connectivity action bursts.

The engine-side interacts with the BPM engine. It receives control events: startup and shutdown calls, service configuration information, deploy events, etc. It also sends the message to the engine for the *Request* operator and receives a message from the engine in the *Response* operator. Because it is invoked by the engine, the engine side speed is limited by the rest of the BPM.

The agent client-side handles all client interaction. It deals with incoming client connections and data arriving on those connections, and it sends the responses to that data back to the clients.

## SOAP

Simple Object Access Protocol (SOAP) is for exchanging information in distributed environments. It provides a common message-passing protocol that uses HTTP to carry messages formatted in Extensible Markup Language (XML). A key objective of SOAP is to provide a way for business applications on the Internet and other distributed environments to talk to each other using XML as the basis for information exchange.

SOAP is the communications protocol used across the wholesale-retail boundary. It is also the communication protocol used between the external applications and the SP Director BPM. In the BSM model, the NP Session Controller BPM initiates and takes down a session with the SP through the SOAP interface.

## TCL

The Tcl agent library is a collection of Tcl extensions that allow you to invoke common functionality from the Tcl-enabled Cisco scripts. The system loads the library at startup.

## Device Agents

This section lists the device-specific agents for the following devices:

- SP Repository
- SP Director
- NP Repository

## SP Repository

- Event
- HTTP
- Scheduler
- Socket
- TCL

## SP Director

- event
- http
- scheduler
- socket
- tcl

## NP Repository

- event
- socket
- tcl

# 1

## Platform Specifications

---

### Overview

This chapter discusses specifications for approved platforms for the Broadband Policy Manager (BPM) device.

Figure 1-1 Sun Server - Sun Fire V20z Specifications.

| ITEM                                                                                              | SPECIFICATION                                                                                          |
|---------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| <b>PROCESSOR</b>                                                                                  |                                                                                                        |
| Processor                                                                                         | Dual 2.8-GHz Intel Xeon processor (533-MHz FSB)                                                        |
| Cache                                                                                             | Up to 1-MB cache for each processor                                                                    |
| <b>MAIN MEMORY</b>                                                                                |                                                                                                        |
|                                                                                                   | 2 GB of registered DDR-266 ECC SDRAM                                                                   |
| <b>STANDARD/INTEGRATED INTERFACES</b>                                                             |                                                                                                        |
| Network                                                                                           | Dual 10/100/1000-Mbps Base-T Ethernet ports                                                            |
| Serial                                                                                            | One RS-232 RJ45 connector (rear accessible)                                                            |
| USB                                                                                               | Three USB ports (one front, two rear)                                                                  |
| PCI                                                                                               | Two 64-bit, 100-MHz PCI-X slots (one full height/length, one low profile)                              |
| SCSI                                                                                              | Adaptec 7902 dual-channel Ultra320 SCSI controller and rear dual channel Ultra320 SCSI VHDCI connector |
| Video                                                                                             | Integrated ATI Rage XL video controller                                                                |
| <b>MASS STORAGE AND MEDIA</b>                                                                     |                                                                                                        |
| Host adapters                                                                                     | Adaptec 7902 dual-channel U320 SCSI controller to support the internal disks and CD-ROM drive          |
| Internal disk                                                                                     | Up to three 3.5-in. x 1.0-in. 36-GB or 73-GB 10K RPM drives available                                  |
| Internal CD-ROM                                                                                   | CD-ROM drive                                                                                           |
| Internal DVD                                                                                      | Combo DVD-ROM/floppy drive available as an option                                                      |
| <b>SOFTWARE</b>                                                                                   |                                                                                                        |
| Operating System                                                                                  | Solaris 9 8/03 Operating System, x86 Platform Edition                                                  |
| <b>CONSOLE OPTIONS</b>                                                                            |                                                                                                        |
| Keyboard and mouse port                                                                           | PS/2 keyboard and mouse                                                                                |
| Video                                                                                             | VGA video connector                                                                                    |
| <b>ENVIRONMENT</b>                                                                                |                                                                                                        |
| Power                                                                                             | 100 V to 240 V, 50/60 Hz, 350 W PFC supply, dual redundant fans                                        |
| Operating temperature                                                                             | 10° C to 35° C (50° F to 95° F)                                                                        |
| Nonoperating temperature                                                                          | -40° C to 70° C (-40° F to 158° F),                                                                    |
| Operating vibration                                                                               | 0.15 G z-axis; 0.10 G x- and y-axis; 5 to 500 Hz sine                                                  |
| Nonoperating Vibration                                                                            | 0.5 G z-axis; 0.25 G x- and y-axis, 5 to 500 Hz sine                                                   |
| Operating Shock                                                                                   | 2 G, 11 ms, half sine                                                                                  |
| <b>REGULATIONS - Meets or exceeds these requirements:</b>                                         |                                                                                                        |
| FCC Class A, CISPR 22A, VCCI-A, UL, CE, Austel, I, GOST R, MIC, Nemko GS, Canada ICES-003 Class A |                                                                                                        |
| <b>DIMENSIONS AND WEIGHTS</b>                                                                     |                                                                                                        |



| ITEM   | SPECIFICATION                                                                                        |
|--------|------------------------------------------------------------------------------------------------------|
| Height | 43 mm (1.69 in.)                                                                                     |
| Width  | 430 mm (16.93 in.)                                                                                   |
| Depth  | 609 mm (23.89 in.)                                                                                   |
| Weight | Unpacked two-drive server: 15.9 kg (35 lb.), approximate;<br>Packaged: 18.1 kg (40 lb.), approximate |
| Rack   | Fits 19-, 23-, 24-inch, and 600 mm racks                                                             |





|                               |                                                                                                                                                                                                                                                                   |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>A</b>                      |                                                                                                                                                                                                                                                                   |
| Active                        | In a pair of BPMs, this is the BPM actively processing requests.                                                                                                                                                                                                  |
| AE                            | Application Environment.                                                                                                                                                                                                                                          |
| Agent                         | An internal PCS component that interacts with a device. The user creates the agent and configures it to interact with a specific device by indicating the device type, IP address, and port number. The user then assigns the agent to perform service functions. |
| Agent Configuration           | Agent information that comprises a specific agent type instance. For example, a RADIUS agent configuration contains appropriate IP address, port, and shared secret values for a RADIUS agent type.                                                               |
| Agent Function                | See Function.                                                                                                                                                                                                                                                     |
| Agent Instance                | A running instance of an agent type.                                                                                                                                                                                                                              |
| Agent Package                 | Software that allows agents to interact with a particular device type. For example, a RADIUS agent package contains software that allows the PCS user to create agents that interact with specific RADIUS devices.                                                |
| Agent Type                    | The agent type describes a particular type of agent that you can load onto the system. You select the agent type when you create the agent instance.                                                                                                              |
| API                           | Application Program Interface.                                                                                                                                                                                                                                    |
| Application                   | A service that maps business models and operational procedures directly into IP services, executable by their customers, for example, video on demand or automatic backup. See also Service.                                                                      |
| Application Program Interface | API.                                                                                                                                                                                                                                                              |
| Application Service Provider  | ASP.                                                                                                                                                                                                                                                              |
| ASP                           | Application Service Provider.                                                                                                                                                                                                                                     |
| Asynchronous Transfer Mode    | ATM.                                                                                                                                                                                                                                                              |
| ATM                           | Asynchronous Transfer Mode.                                                                                                                                                                                                                                       |
| Attribute                     | In the BPDS Object manager tool, a simple type with a default value. An object can have several attributes.                                                                                                                                                       |
| <b>B</b>                      |                                                                                                                                                                                                                                                                   |
| Backend                       | Software that runs on the BPM. It comprises the controller, engine, agent host, activation daemon, and scheduler processes; synonymous with BPM.                                                                                                                  |

|                                 |                                                                                                                                                                                                                            |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BGP                             | Border Gateway Protocol.                                                                                                                                                                                                   |
| Border Gateway Protocol         | BGP. An exterior gateway routing protocol that enables groups of routers to share routing information to establish efficient, loop-free routes. BGP is commonly used within and between Internet Service Providers (ISPs). |
| BPDS                            | Broadband Policy Design Studio.                                                                                                                                                                                            |
| BRAS                            | Broadband Remote Access Server (10K device).                                                                                                                                                                               |
| Broadband Policy Design Studio. | BPDS.                                                                                                                                                                                                                      |
| Broadband Policy Manager        | BPM.                                                                                                                                                                                                                       |
| Broadband IP Services Module    | BSM.                                                                                                                                                                                                                       |
| Broadband Remote Access Server  | BRAS. (10K device)                                                                                                                                                                                                         |
| BSM                             | Broadband IP Services Module                                                                                                                                                                                               |
| <b>C</b>                        |                                                                                                                                                                                                                            |
| Class of Service                | CoS. A traffic prioritization scheme that enables more predictable traffic delivery, based on application requirements.                                                                                                    |
| Client                          | Generic term that denotes the BPM BPDS application.                                                                                                                                                                        |
| CoS                             | Class of Service.                                                                                                                                                                                                          |
| CPE                             | customer premises equipment.                                                                                                                                                                                               |
| CLI                             | command line interface.                                                                                                                                                                                                    |
| Cluster                         | A pair of cooperating and redundant BPMs.                                                                                                                                                                                  |
| Component                       | An object comprising data and code. A component provides a well-specified set of publicly available services. All devices, services, and applications on a network are components.                                         |
| Configuration                   | Information necessary to construct an instance of a type (agent, service).                                                                                                                                                 |
| Controller                      | A software element that runs on the BPM and controls various elements of the backend. Usually only one controller exists per backend; therefore, from the BPDS perspective, the controller is the backend.                 |
| Customer Premises Equipment     | CPE.                                                                                                                                                                                                                       |
| <b>D</b>                        |                                                                                                                                                                                                                            |
| Deep Packet Inspection Protocol | DPI.                                                                                                                                                                                                                       |
| Device                          | Any piece of software or hardware connected to a network. RADIUS servers, routers, billing systems, accounting systems, and video servers are devices. An agent communicates with a device.                                |

|                        |                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Device Rule            | A generic task that can be applied to a device that supports the device rule functionality to carry out a network action. Device rules can retrieve information from connected devices. See also Device Type and Policy Rule.                          |
| Device Type            | A classification of actual devices, based on device attributes, such as vendor, model, hardware version, and software version. See also Device Rule.                                                                                                   |
| Domain                 | One or more cooperating Policy Control System (PCS)s managed by a single domain repository.                                                                                                                                                            |
| Domain Controller      | The software that stores and manages agents, agent types, agent instances, service types, services instances packages, service engines, policy engines, and clusters; also known as the Cisco database.                                                |
| Domain Data            | Data maintained about the elements in a domain; for example, controller host and port configuration, database host and port information, agent and service configuration and deployment information.                                                   |
| Domain Repository      | The master database that contains configuration information for each domain element.                                                                                                                                                                   |
| DPI                    | Deep Packet Inspection Protocol.                                                                                                                                                                                                                       |
| DSLAM                  | Digital Subscriber Line Access Multiplexer.                                                                                                                                                                                                            |
| <b>E</b>               |                                                                                                                                                                                                                                                        |
| Element                | An object with the BPM: Cisco package; agent configuration; service instance; shared object.                                                                                                                                                           |
| Enumeration            | In the BPDS Object manager tool, enumeration is contained within a simple type.                                                                                                                                                                        |
| <b>F</b>               |                                                                                                                                                                                                                                                        |
| Field Replaceable Unit | FRU. An FRU represents an element (e.g., entire system, BPDS client software, agent) within the Policy Control System (PCS) that can have a version associated with it. A FRU is a subset of an element.                                               |
| Flow                   | The movement of data or control between agents. It is a collection of one or more operators and zero or more routes. The designer uses flows to define services and applications.                                                                      |
| FRU                    | Field Replaceable Unit.                                                                                                                                                                                                                                |
| Function               | The element that performs an operation, based on inputs and returns the results of the operation via its outputs. The user drags and drops a function into a flow in the BPDS. An agent, which interacts with a device, performs the actual operation. |
| <b>G</b>               |                                                                                                                                                                                                                                                        |
| GUI                    | Graphical User Interface (BPDS).                                                                                                                                                                                                                       |
| <b>H</b>               |                                                                                                                                                                                                                                                        |
| Head Version           | The latest version of an element.                                                                                                                                                                                                                      |
| Hot Backup             | A backup system, identical in configuration to the primary BPM system. The hot backup is ready to resume request execution immediately if the primary BPM system fails. See Primary BPM, Secondary BPM.                                                |
| <b>I</b>               |                                                                                                                                                                                                                                                        |
| ILC                    | Appium SOAP load balancer.                                                                                                                                                                                                                             |
| Implementation         | An instruction set for executing a specification.                                                                                                                                                                                                      |

|                               |                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Install Rule                  |                                                                                                                                                                                                                                                                                                                                                       |
| Instance                      | An executing type (agent, service), created from a specification, implementation, and configuration. An agent instance is a specific implementation of that agent type.                                                                                                                                                                               |
| Interface                     | A collection of functions.                                                                                                                                                                                                                                                                                                                            |
| IP address                    | The address that identifies a computer. The IP address format is a 32-bit numeric address written as four numbers (0 to 255) separated by periods.                                                                                                                                                                                                    |
| ISP                           | Internet Service Provider.                                                                                                                                                                                                                                                                                                                            |
| <b>J</b>                      |                                                                                                                                                                                                                                                                                                                                                       |
| JDBC™                         | Java Database Connectivity (TM of Sun Microsystems; not an acronym).                                                                                                                                                                                                                                                                                  |
| <b>L</b>                      |                                                                                                                                                                                                                                                                                                                                                       |
| L2TP                          | Layer Two (2) Tunneling Protocol.                                                                                                                                                                                                                                                                                                                     |
| <b>M</b>                      |                                                                                                                                                                                                                                                                                                                                                       |
| Management Protocol           | A set of protocols for managing devices and networks, for example SNMP, SSH.                                                                                                                                                                                                                                                                          |
| Master                        | In a pair of BPMs, the BPM that actively processes requests. Also known as the primary BPM.                                                                                                                                                                                                                                                           |
| Metadata                      | In the BPDS Object Manager tool, the data structure. Cisco customers can import metadata to invoke a structure for their database.                                                                                                                                                                                                                    |
| MPLS                          | Multiprotocol Label Switching.                                                                                                                                                                                                                                                                                                                        |
| Multiprotocol Label Switching | An IETF initiative that integrates Layer 2 information about network links (bandwidth, latency, utilization) into Layer 3 (IP) within a particular autonomous system or ISP. It simplifies and improves IP-packet exchange and gives network operators the flexibility to divert and route traffic around link failures, congestion, and bottlenecks. |
| <b>N</b>                      |                                                                                                                                                                                                                                                                                                                                                       |
| N + 1 Redundancy              | The ability for service engines to use one service engine as a backup.                                                                                                                                                                                                                                                                                |
| NAS                           | Network Attached Storage.                                                                                                                                                                                                                                                                                                                             |
| NAV                           | The Network Admin view of the PCS VDS.                                                                                                                                                                                                                                                                                                                |
| Network                       | Loosely coupled set of components.                                                                                                                                                                                                                                                                                                                    |
| Network Attached Storage      | NAS.                                                                                                                                                                                                                                                                                                                                                  |
| Network Device Library Entry  | An entry that properly configures a network device for applying a network policy.                                                                                                                                                                                                                                                                     |
| Network Event                 | Contained within a profile, a set of install and uninstall rules performed in sequence.                                                                                                                                                                                                                                                               |
| Network Policy                | A device rule entry, where the device rule contains commands to configure a network device to apply a network policy. See also Device Rule, Policy Rule.                                                                                                                                                                                              |

|                                        |                                                                                                                                                                                                                                                           |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network Provisioner                    | A Cisco system on the wholesale side of a wholesale/retail network.                                                                                                                                                                                       |
| NP                                     | Network Provider.                                                                                                                                                                                                                                         |
| <b>O</b>                               |                                                                                                                                                                                                                                                           |
| Object                                 | An agent, controller, function, service, switch, or service within the Policy Control System (PCS).                                                                                                                                                       |
| Object Dependency                      | An exact object type, for example a Cisco 2500 router agent, that a service depends on. The user adds the object type to the dependency list of the service. All Interfaces supported by the object type are then available for use with the service.     |
| Object Type                            | In the BPDS Object manager tool, object type is defined with attributes. It can own contain, and associate with other object types.                                                                                                                       |
| OC                                     | Orchestration Controller. That portion of the Policy Control System (PCS) that controls processes such as username and password authentication,                                                                                                           |
| Operation and Support Systems          | OSS.                                                                                                                                                                                                                                                      |
| Operator                               | A representation of actions to be undertaken on a system networked to a Policy Control System (PCS).                                                                                                                                                      |
| Orchestration Controller               | OC. See Domain Controller.                                                                                                                                                                                                                                |
| Orchestration Network                  | The process for handling service calls over a network. It defines the flow of control and information between work units.                                                                                                                                 |
| OSS                                    | Operation and Support Systems.                                                                                                                                                                                                                            |
| <b>P</b>                               |                                                                                                                                                                                                                                                           |
| Pad                                    | A collection of pins on an operator. This appears as a box along the edge of an operator.                                                                                                                                                                 |
| BPM                                    | Broadband Policy Manager.                                                                                                                                                                                                                                 |
| PDP                                    | Policy Decision Point.                                                                                                                                                                                                                                    |
| PE                                     | Policy Engine. The software that stores and manages user profile information, subscriber access records, policy rules; also known as the policy database.                                                                                                 |
| PEP                                    | Policy Enforcement Point.                                                                                                                                                                                                                                 |
| Pin                                    | An input or output from an operator. The pin serves as a route endpoint and holds a single input or output value. For example, an operator that needs a username and password as input has two input pins; one for the username; the other, the password. |
| PM                                     | Policy Manager.                                                                                                                                                                                                                                           |
| Point-to-Point Protocol Over ATM       | PPPoA.                                                                                                                                                                                                                                                    |
| Point-to-Point Termination Aggregation | PTA.                                                                                                                                                                                                                                                      |

|                                            |                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Policy                                     | A flow comprising a rule or set of rules that take a specific action provided by an ISP for its users. For example, a policy for subscriber access directs how the system identifies a user via user id, access type, and log in location. A policy performs an operation, based on input and returns the results of its action as output.                                                              |
| Policy Database                            | The database of policy objects that services access to make policy decisions.                                                                                                                                                                                                                                                                                                                           |
| Policy Decision Point                      | PDP.                                                                                                                                                                                                                                                                                                                                                                                                    |
| Policy Enforcement Point                   | PEP.                                                                                                                                                                                                                                                                                                                                                                                                    |
| Policy Engine                              | PE. The software that stores and manages user profile information, subscriber access records, policy rules; also known as the policy database.                                                                                                                                                                                                                                                          |
| Policy Function                            | Policy rules encapsulated in a TCL agent <i>execute</i> function.                                                                                                                                                                                                                                                                                                                                       |
| PPPoA                                      | Point-to-Point Protocol Over Asynchronous Transfer Mode.                                                                                                                                                                                                                                                                                                                                                |
| PR                                         | Policy Repository.                                                                                                                                                                                                                                                                                                                                                                                      |
| Primary BPM                                | In a pair of BPMs, the primary BPM is processing requests. If the primary fails, the secondary becomes the primary.                                                                                                                                                                                                                                                                                     |
| Profile                                    | A sequence of network events involved in creating and taking down a network session for a subscriber.                                                                                                                                                                                                                                                                                                   |
| Property                                   | The parameter or characteristic of an agent or device.                                                                                                                                                                                                                                                                                                                                                  |
| PTA                                        | Point-to-Point Termination Aggregation.                                                                                                                                                                                                                                                                                                                                                                 |
| <b>R</b>                                   |                                                                                                                                                                                                                                                                                                                                                                                                         |
| RADIUS                                     | Remote Authentication Dial-In User Service.                                                                                                                                                                                                                                                                                                                                                             |
| Remote Authentication Dial-in User Service | RADIUS. A client/server protocol enabling remote access server communication with a central server to authenticate dial-in users and authorize their access to the requested system or service. RADIUS allows a company to maintain user profiles in a central database. It provides better security, allowing a company to set up a policy that can be applied at a single administered network point. |
| Remote Method Invocation                   | (RMI)                                                                                                                                                                                                                                                                                                                                                                                                   |
| Retail                                     | The Service Provider side of a network.                                                                                                                                                                                                                                                                                                                                                                 |
| RMI                                        | Remote Method Invocation                                                                                                                                                                                                                                                                                                                                                                                |
| Role-based Dependency                      | A dependency in which a user indicates that multiple service elements support the same interface. The user defines different roles and assigns the required service interfaces to each. The different roles are added to the dependency list for the service and operators are clearly marked to indicate their assigned role.                                                                          |
| Route                                      | A path between operators.                                                                                                                                                                                                                                                                                                                                                                               |
| RS                                         | Remote Server.                                                                                                                                                                                                                                                                                                                                                                                          |
| Rule                                       | Criteria applied to the objects and methods of a business system to determine how objects and methods are used by, or for, a given system user. A flow comprises a rule or set of rules. Rules prescribe terms and conditions for a specific action provided by an ISP for its end users. One rule can call another rule.                                                                               |



| S                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SAV                                | The Service Admin view of the BPM BPDS.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Schema                             | A set of rules and syntax for storing data.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| SDV                                | The Service Design view of the BPM BPDS.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| SE                                 | Service Engine.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Secondary BPM                      | In a pair of BPMs, the secondary BPM does not process requests. Instead, it monitors the health of the primary to assess its ability to process requests. If the primary is not viable, the secondary becomes the primary.                                                                                                                                                                                                                                    |
| Service                            | An application, created by the PCS user, that maps business models and operational procedures directly into IP services, executable by their customers, for example, video on demand or automatic backup. A service comprises objects (agent, controller, function, switch, or other service). A service comprises one or more flows.                                                                                                                         |
| Service Configuration              | The information needed to construct a service. The service configuration specifies agent configurations for each function in the service type. The BPM user creates the service configuration.                                                                                                                                                                                                                                                                |
| Service Dependency                 | The user-defined dependencies of a service. The user builds a service by defining data-flows that use operators from multiple objects, including agents and other services. The user builds a service upon a concrete set of agents and services.<br><br>If a service is portable across different agents and services, the user specifies any constraints on the concrete instances and specifies the interfaces that those concrete instances must support. |
| Service Engine                     | SE. The software that runs services; also known as the backend.                                                                                                                                                                                                                                                                                                                                                                                               |
| Service Interface Dependency       | If a service uses a particular service interface, but does not require that a specific object provide the service interface, the user can add the service interface as a dependency. Here, the service interface operators are available for use in the current service, but the object that provides the interface is determined later.                                                                                                                      |
| Service Instance                   | The running of a service type created by the end user.                                                                                                                                                                                                                                                                                                                                                                                                        |
| Service Palette                    | The agent types available to a service.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Service Profile                    | A collection of services and information about service execution.                                                                                                                                                                                                                                                                                                                                                                                             |
| Service Provisioner                | SP. A Cisco appliance on the retail side of a wholesale/retail network.                                                                                                                                                                                                                                                                                                                                                                                       |
| Service Type                       | The definition of what agent types are required for a service; the defined flow of data between functions of agent types. The end user or PCS staff creates the service type.                                                                                                                                                                                                                                                                                 |
| Servlet                            | An applet that runs on a server. Usually refers to a Java applet that runs within a Web server environment. Analogous to a Java applet that runs within a Web browser environment.                                                                                                                                                                                                                                                                            |
| Shared Secret                      | An authentication string that ensures security between devices. KERBEROS is an instance of a shared-secret authentication protocol.                                                                                                                                                                                                                                                                                                                           |
| Simple Type                        | In the BPDS Object manager tool, a simple type is similar to data type, except it can express with enumerations.                                                                                                                                                                                                                                                                                                                                              |
| Simple Network Management Protocol | SNMP. An IETF protocol by which networked devices can be periodically polled for information as part of a network management system.                                                                                                                                                                                                                                                                                                                          |

|                           |                                                                                                                                                            |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLA                       | Service Level Agreement.                                                                                                                                   |
| SNMP                      | Simple Network Management Protocol.                                                                                                                        |
| SOAP                      | Simple Object Access Protocol.                                                                                                                             |
| SP                        | Service Provider.                                                                                                                                          |
| SPACE-P                   | Space Provisioning.                                                                                                                                        |
| Specification             | A type definition that includes interface definitions, configuration schemas, and binding information.                                                     |
| SQL                       | Structured Query Language.                                                                                                                                 |
| Structured Query Language | SQL.                                                                                                                                                       |
| Subscriber                | A customer of a business that provides a wide variety of online services, including e-mail, stock quotes, news, and online forums.                         |
| Subscriber Profile        | A table entry containing information, such as authentication, authorization, and location on a specific subscriber.                                        |
| Super Operator            | A reusable flow that other flows can call. To the other flows, the super operator appears as an operator that they can call and insert on any route.       |
| Switch                    | A device that filters and forwards packets between LAN segments. Switches operate at the data link layer and the network layer of the OSI Reference Model. |

**T**

|                    |                                                                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| TER                | Transaction Event Record.                                                                                                                       |
| TM                 | Traffic Management (TM).                                                                                                                        |
| TOS                | Type of Service.                                                                                                                                |
| Traffic Management | TM.                                                                                                                                             |
| TRPC               | Transaction Remote Procedure Call.                                                                                                              |
| Type               | A PCS component group that has a unique specification. It may have an implementation, and it may have one or more configurations and instances. |

**U**

|     |                                                                                                                            |
|-----|----------------------------------------------------------------------------------------------------------------------------|
| UML | Unified Modeling Language. UML is the industry standard notation for representing software architecture and design models. |
|-----|----------------------------------------------------------------------------------------------------------------------------|

**V**

|      |                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------|
| VC   | Virtual Circuit.                                                                                               |
| BPDS | The graphical user interface administering the Cisco BPM. The application for creating and deploying services. |

|               |                                                                                                                                                                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VLAN          | Virtual LAN. A network of computers that behave as if they are connected to the same wire even though they may be physically located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible. |
| Voice-over-IP | VoIP. Voice delivered using the Internet Protocol.                                                                                                                                                                                                                    |
| VoIP          | Voice-over-IP.                                                                                                                                                                                                                                                        |
| VP            | Virtual Path.                                                                                                                                                                                                                                                         |
| VPN           | Virtual Private Network.                                                                                                                                                                                                                                              |
| <b>W</b>      |                                                                                                                                                                                                                                                                       |
| WDSL          | Wireless Digital Subscriber Line.                                                                                                                                                                                                                                     |
| Workspace     | The BPDS area where the designer visually programs services.                                                                                                                                                                                                          |
| Wholesale     | The Network Provider side of a network.                                                                                                                                                                                                                               |
| WSA           | Web Service Adaptor.                                                                                                                                                                                                                                                  |



---

## Symbols

3, 16

5

---

## Numerics

01\_adminServiceTrigger 3

03\_refreshSCPCConfig 3

10K applications 1

10K BRAS device 2, 6, 9, 11, 2

20\_adminMgmt 3

---

## A

A 4

access

methods 10, 13, 15, 18

    PPPoA in L2TP Mode 12, 15

    PPPoA in PTA or LAC/LNS Mode 12, 15

    PPPoA in RA-MPLS Mode 12, 15

    PPPoE in L2TP Mode 12, 15

    PPPoE in PTA or LAC/LNS Mode 12, 15

    PPPoE in RA-MPLS Mode 12, 15

    PPPoEoA in L2TP Mode 12, 15

    PPPoEoA in PTA or LAC/LNS Mode 12, 15

    PPPoEoA in RA-MPLS Mode 12, 15

parental control of 16

accounting

    basic

        RADIUS 9

    basic RADIUS 6, 12

    interim 14, 2

    start 7, 14, 2, 6, 9, 11, 2

    stop 14, 15, 2

activating

    breach services 8

    bundles 7

    services 8

- agents
  - device
    - Retail Director RPC
      - Event 4
      - HTTP 4
      - Scheduler 4
      - Socket 4
      - TCL 4
    - Retail Repository
      - Event 3
      - HTTP 3
      - Scheduler 3
      - Socket 3
      - TCL 3
    - Wholesale Repository
      - Event 4
      - Socket 4
      - TCL 4
  - and 14
  - Application Firewall (Family Control-Oriented) application 3, 8, 16
  - Application layer 4
  - applications 11
    - 10K 1
    - and BSM components 3
    - and service requests 3
    - Application Firewall (Family Control-Oriented) 1, 3, 8, 16
    - Bandwidth on Demand 1, 2
    - Bandwidth-on-Demand 1, 2
    - BSM 1
    - control-oriented 3, 16
    - developing 19
    - Family Control-Oriented application 1, 16
    - functions of 4
    - Layer 7 13
    - quota-based, tiered 11
    - Quota-Based Tiered 1, 2, 11
    - Quota-Based Tiered Applications 8, 11
    - registered 15
    - schema
      - installing on server 8
    - Tiered 8

- tiered [8, 11](#)
- tiered applications [1, 2](#)
- Time-Based Turbo Bandwidth [1, 2, 1, 5](#)
- VoIP-Oriented Preferential [1, 3, 14](#)
- VoIP-Oriented Preferential Applications [8, 14](#)
- ApplyPolicy [5](#)
- authentication [2](#)

---

## B

- bandwidth [2, 6, 9, 10, 12](#)
- Bandwidth on Demand [2](#)
- Bandwidth on Demand application [1, 2, 1, 2](#)
- blocking service [11, 17](#)
- BoD [2](#)
- Boost package support [9](#)
- BPM Director [3, 4](#)
- BPM entries
  - network [1](#)
  - service [1](#)
- BPM Policy Repository [3](#)
- BPM Session Controller [3](#)
- breaches
  - activating service for [7](#)
  - and quota limits [7](#)
  - and quota reset [10](#)
  - and use of network profile [7](#)
  - detecting [10](#)
  - exiting from [10](#)
  - of quota [2, 11](#)
  - quota [8, 9, 13](#)
  - service [8](#)
  - subscriber condition [2, 11](#)
  - tracking [9](#)
    - service [9](#)
- breach services
  - activating [8](#)
  - containing event trigger [10](#)
  - running when breach detected [10](#)

- bronze service level [2, 8, 11](#)
- bronze tiered package [8](#)
- BSM
  - components [3](#)
  - service requests [3](#)
- BSM applications [1](#)
- bundle events [10](#)
- bundles
  - activating [7](#)
  - and service events [9, 10](#)
  - and subscriber data treatment types [10](#)
  - service [8, 9, 10](#)
    - and volume supplement limits [10](#)
  - service event [10](#)
  - service event in [6](#)
  - service events in [9](#)

---

## C

- catalogs [10](#)
  - dynamic service [2, 4, 6, 7, 9, 13, 15](#)
- command line interface (CLI) [5](#)
- components [3](#)
  - network [3](#)
- configuration [4](#)
- configurations
  - network [10](#)
  - service [10](#)
- contracts
  - subscriber [2, 5](#)
  - terms of [2, 5](#)
- control-oriented applications [3, 16](#)
- conventions
  - text [2](#)
  - used in guide [1](#)
- creating
  - quotas
    - supplemental [10](#)
  - supplemental quota [10](#)



---

**D**

## Datacaster

- command line interface (CLI) 5

## detecting

- breaches 10

## developing

- applications 19

## devices

## agents

- Event 3, 4

- HTTP 3, 4

- Scheduler 3, 4

- Socket 3, 4

- TCL 3, 4

- management protocols for 12

Director 4, 1

director 3

domain controller 3

Downstream/Upstream/Aggregate 13

dynamic configuration 2

dynamic service 10

---

**E**

email 8

## entries

- BPM configuration type 8

- configuration 7

- Director Service 15

- NAS Director 15

- network BPM 1, 8

- NP Director BPM 15

- remote data store 9, 10

- service BPM 8

- Service Directory 4

- type 1

Ethernet 2, 6, 9, 11

event interfaces

- SOAP 7
- events
  - bundle 10
  - level 9, 10
  - quota 9
  - quota threshold 8
  - service 9, 10
    - level 10
    - quota 9
  - trigger 10
- event triggers
  - and breach service 10
- Extensible Markup Language (XML) 3

---

## F

- failed server
  - recovery from 8
- Family Control-Oriented application 3, 8, 16
- files
  - secureAccessDatabaseGUISupport.tzz 7, 4
- File Transfer Protocol 9
- firewall 16
- flows
  - radiusStartAcct 15
  - radiusStopAccountService 15
- FTP 9

---

## G

- generateStatisticFilesPR 3
- gold service level 2, 8, 11
- graphical user interface (GUI) 1
- groups
  - service 9
- GUI 1

---

## H

HyperText Transfer Protocol (HTTP) 7, 3

---

## I

identifier mappings 5  
identifiers  
    contents of 14  
    subscriber 14  
    unique for subscriber 14  
implementation 3  
interfaces  
    SOAP 14, 4, 7, 10, 13, 15, 18  
interim  
    accounting 2  
Internet service levels 2, 8  
Internet with VoIP 14  
IP services  
    subscriber-based 1  
ISP 4

---

## L

layer 5  
Layer 7 applications 13  
layers  
    Application 4  
levels  
    service event 10  
limits 9  
    on service consumption 9  
    on service iterations 9  
    service usage 9  
    usage duration 9  
local table 5  
logic  
    quota-based 13  
logs 5

---

**M**

mappings

identifier 5

MPLS 3, 4

MPLS (Multiprotocol Label Switching) 3, 7, 9, 12, 15, 17

Multiprotocol Label Switching (MPLS) 3, 7, 9, 12, 15, 17, 4

---

**N**

network 4

and subscriber identity 14

and subscriber logon 9, 10

and unique identifier 5

configuration 10

device 9, 10

devices 15

hardware 20

policies 4, 15

profile 8, 9, 10

resources 10, 14, 20

sessions 5

subscriber identification on 14

wholesale 13

network BPM entries 1

network components 3

network policies

activating 4

deactivating 4

networks

and subscriber sessions 14

components 3

retail 3

transport 4

wholesale 3

notifyAccessLineChange 3

NP 2

---

**O**

- object 5
- option
  - quota-based 5

---

**P**

- parental control 16
- password 5, 14, 9
- platform specifications 1
- policies 4, 5
- policy database 2, 6, 9, 11
- policy management 1
- Policy Manager 1
- policy repository 3, 4, 10, 13, 14, 1, 2, 3, 4, 7, 8, 9, 10, 3, 9
- Policy Repository BPM 3
- PPP client 2, 6, 9, 11
- processBPMRegistration 4
- processnetworkCacheRequest 3
- processServiceCacheRequest 4
- processSession ControllerRegistration 3
- profiles
  - service 9
- protocols
  - File Transfer Protocol 9
  - FTP 9
  - HTTP 7, 3
  - MPLS 3, 7, 9, 12, 15, 17, 4
  - Multiprotocol Label Switching (MPLS) 3, 4
  - RADIUS 4, 6, 9, 12, 1, 6
  - Remote Access Dial-In Service (RADIUS) 4, 6, 9, 12, 1, 6
  - Remote Method Invocation (RMI) 7, 6
  - RMI 7, 6
  - Simple Network Management Protocol (SNMP) 3, 7, 10, 12, 15, 17, 4, 7, 8
  - SNMP 3, 7, 10, 12, 15, 17, 4, 7, 8
  - SOAP 7, 14, 3, 4, 7, 10, 12, 13, 15, 17, 18, 3
  - TRPC 5, 10, 14, 8
  - XML 3

Purchase 13

---

## Q

quota 2, 11, 12

breach of 8

increasing limit of 8

purchasing 8

service event 6

specifying network profile for breach 7

supplement 8

triggers 10

Quota Assignment per application 13

quota-based

logic 13

option 5

tiered application 11

quota-based, tiered 11

Quota-Based Tiered Applications 8, 11

Quota-Based Tiered applications 1, 2, 11

quota-based tiered applications 11

quota management 13

quota resets

and breaches 10

Quota Reset Timer 13

quotas 8

and subscriber breach 10

breach 9

duration 10

resetting 10

service event 9, 7

service events 9

supplement

creating 10

supplemental

bundle level 10

service event level 10

volume 10

tracking a breach 9

tracking a breach with 9

---

## R

RADIUS 4, 6, 9, 12, 2, 1, 6

- accounting 6, 9, 12
  - start 14
  - stop 14
- accounting stop 15
- agent 1
- authentication 14, 4
- radiusStartAcct 15
- radiusStopAccountService 15
- recovery
  - after failed server 8
- registered application 15
- registerPCS 7
- Remote Access Dial-In Service (RADIUS) 4, 6, 9, 12, 1, 6
- Remote Authentication Dial-In User Service (RADIUS) 2, 6
- Remote Method Invocation (RMI) 7, 6
- requests
  - BSM service 3
- Retail Director RPC
  - agents for 3
- retail domain 1, 3, 4, 5, 10, 2
- retail domain components 4
- Retail Repository
  - agents for 3
- RMI 7, 6
- route table 4
- rules 5

---

## S

secureAccess 7

secureAccessDatabaseGUISupport.tzz 7, 4

secureAccessRegistration 7

secureAccessRegistration table 7

- secureAccessRegistration tables 7
- secureAccess tables 7
- secureAccessUsage 7
- secureAccessUsage tables 7
- service 1
  - configuration 10, 7
  - type 7
- service BPM entries 1
- service bundles 8, 9, 10, 6, 14
  - and volume supplement limits 10
- service consumption
  - limits 9
- Service Director Entries
  - table 4
- serviceDirectorEntry 7
- serviceDirectorEntry table 7
- Service Directory
  - entries 4
- Service Directory Entry table 4
- service event 6
- service events 9, 10, 6
  - aggregated into service profiles 6
  - and bundles 9
  - and scheduled services 6
  - and service profile 6
  - in bundle 9
  - level 10
  - parameters defined in 6
  - quota 9, 6, 7
  - quotas attached to 6
  - running service for subscriber 6
  - scheduled 6
  - scheduling service execution 6
  - support for 6
  - timed 4, 6, 7
- service group
  - containing subscribers 14
- service groups 9
  - and associated profile 14



- and associated service profile 14
- and domains 14
- and override by service profile 14
- and override by subscriber profile 14
- and primary accounts 14
- and profiles 13
- and service bundle 14
- and service profile 14
- and service profiles 6, 14
- and subaccounts 14
- and subscribers 14
- and subscriber services 14
- benefits of 6
- domain assignment 6
- profile referenced by 13
- service profiles associated with 6
- services available to 6
- subscriber 9, 6
- use of 4
- without associated service profile 6
- without associated service profiles 14
- without profiles 14

service iterations

- limits on 9

service level 11

service level bandwidth 2, 6, 9

service levels

- bronze 2, 8, 11
- gold 2, 8, 11
- Internet 2, 8
- silver 2, 8, 11

service package definitions 5

service packages 5

service profile 6, 7, 8, 14, 15, 4, 6, 9, 12, 15

serviceProfileEventMgmt 3

service profiles 9

serviceProfileSessionUpdate 3

service provider 1, 2, 3, 2, 5, 8, 14, 16

service providers 3, 4, 9, 17, 19

- service requests
  - and applications 3
- services
  - 01\_adminServiceTrigger 3
  - 03\_refreshSCPCConfig 3
  - 20\_adminMgmt 3
  - activating 8
  - breach 8
  - breaches
    - and supplements 10
  - consumption limit 9
  - consumption limits 9
  - generateStatisticFilesPR 3
  - notifyAccessLineChange 3
  - processBPMRegistration 4
  - processnetworkCacheRequest 3
  - processServiceCacheRequest 4
  - running when breach is detected 10
  - scheduled 10
  - serviceProfileEventMgmt 3
  - serviceProfileSessionUpdate 3
  - spProvisionHttpNotification 4
  - spProvisionMissingAccessline 4
  - spProvisionServiceProfileChange 4
  - spProvisionSIDQuery 4
  - subscriber 9
  - time-based 5
  - tracking breach for 9
  - usage duration 9
- Session Controller 4, 5, 10, 13, 14
- session controller 3
- Session Controller BPM 1, 2, 5, 6, 7, 9, 10
- Session Controllers 4, 5
- silverservice level 2, 8, 11
- Simple Network Management Protocol (SNMP) 3, 7, 10, 12, 15, 17, 4, 7, 8
- Simple Object Access Protocol (SOAP) 7, 14, 3, 4, 7, 10, 12, 13, 15, 17, 18, 3
- SNMP 3, 7, 10, 12, 15, 17, 4, 7, 8
- SP 1, 3
- SP Policy Repository Platform 3

- spProvisionHttpNotification 4
- spProvisionMissingAccessline 4
- spProvisionServiceProfileChange 4
- spProvisionSIDQuery 4
- start
  - accounting 2, 6, 11, 2
- start, accounting 9
- statistics 5
- statistics table 10
- stop
  - accounting 2
- subaccount 17, 18
- subscribe 7
- Subscriber 10
- subscriber 4, 5, 8, 9
- Subscriber Authentication 13
- subscriber-based IP services 1
- subscriber contract 2, 5
- subscriber data 10
- subscriber identifier
  - identifiers
    - subscriber 5
  - unique 14
- subscriber identity 15
- subscriber information 9, 14, 3
- subscriber login 9, 18
- subscriber network profile 2, 6, 9, 11
- subscriber quota 9, 11
- subscriber service 6, 14
- Subscriber Service Change 10, 13
- subscriber service groups 9, 14
  - service profiles associated with 6
- subscriber services 8, 9
  - services
    - subscriber 9
- subscriber session 4, 10, 14, 12, 3, 6, 9
- subscriber usage 10, 14, 15
- supplement 8
- supplemental quotas 10

creating 10

---

## T

table

local 5

tables

accessline 7

route 4

routing 8

secureAccess 7

secureAccessRegistration 7

secureAccessUsage 7

Service Director Entry 4

serviceDirectorEntry 7

table 7

Service Directory Entry 4

subscriber 7

usage 10

usage statistics 10

terms 2, 5

contract 2, 5

text conventions 2

Tiered Applications 8

tiered applications 8, 11

quota-based 11

time-based services 5

time-based turbo 5

Time-Based Turbo Bandwidth application 1, 2, 1, 5

timers

quota reset 13

traffic 1, 8, 11, 2

traffic class 1

Transaction Remote Procedure Call 5

Transaction Remote Procedure Call (TRPC) 5, 10, 14, 8

triggers 2, 7, 10, 12, 15

and wholesale SOAP interface 4

application 7

generic 3

quota [10, 12](#)  
service [3, 4, 10, 12, 15, 17, 18](#)  
TRPC (Transaction Remote Procedure Call) [5, 10, 14, 8](#)  
turbo [5, 6](#)

---

## U

unique [5](#)  
unique subscriber identifier [14](#)  
URLs  
    default  
        for network BPM [1](#)  
    default for service BPM [1](#)  
    fields  
        network [1](#)  
        service [1](#)  
    for network BPM [1](#)  
    for service BPM [1](#)  
        processNtwkEvent/processNtwkEvent [1](#)  
        processServiceEvent/processServiceEvent [1](#)  
    processNtwkEventDirector/processNtwkEventDirector [1](#)  
    processServiceEventDirector/processServiceEventDirector [1](#)  
    processSession ControllerRegistration [3](#)  
usage duration  
    limits  
        service consumption [9](#)  
usage statistics table [10](#)  
username [5, 14, 15](#)

---

## V

video [6](#)  
    on demand [19](#)  
VoIP [9, 14](#)  
VoIP-Oriented Preferential Applications [8, 14](#)  
VoIP-Oriented Preferential applications [1, 3, 14](#)  
Vonage [14](#)

---

**W**

wholesale [2](#)

wholesale domain [13, 14, 8](#)

Wholesale Repository

agents for [3](#)

---

**X**

XML [3](#)