

# PortaSwitch

## Architecture and Concepts



## Copyright Notice & Disclaimers

Copyright © 2000–2017 PortaOne, Inc. All rights reserved

**PortaSwitch Architecture and Concepts, October 2017**

**Maintenance Release 65**

**V 1.65.05**

Please address your comments and suggestions to: Sales Department,  
PortaOne, Inc. Suite #408, 2963 Glen Drive, Coquitlam BC V3B 2P7  
Canada.

Changes may be made periodically to the information in this publication. The changes will be incorporated in new editions of the guide. The software described in this document is furnished under a license agreement, and may be used or copied only in accordance with the terms thereof. It is against the law to copy the software on any other medium, except as specifically provided for in the license agreement. The licensee may make one copy of the software for backup purposes. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopied, recorded or otherwise, without the prior written permission of PortaOne Inc.

The software license and limited warranty for the accompanying products are set forth in the information packet supplied with the product, and are incorporated herein by this reference. If you cannot locate the software license, contact your PortaOne representative for a copy.

All product names mentioned in this manual are for identification purposes only, and are either trademarks or registered trademarks of their respective owners.

## Table of Contents

Preface .....	4
Hardware and Software Requirements .....	5
Disk Space Requirements .....	5
Installation .....	8
What is New in Maintenance Release 64 .....	9
<b>1. System Architecture .....</b>	<b>10</b>
Overview .....	11
Centralized Configuration Management .....	12
Per-configuration Licensing: Flexibility and Control .....	17
Deploying PortaSwitch® Across Multiple Sites .....	20
PortaSwitch® Application Components .....	28
PortaOne Monitoring System .....	32
Updating the System to a New Version .....	37
Zero-Downtime Update .....	40
Custom Modification Management .....	42
PortaSwitch® Deployment in a Cloud .....	43
<b>2. Integration with Third-Party Systems .....</b>	<b>46</b>
Overview .....	47
API for Data Operations .....	47
API for Computer Telephony Integration Services .....	48
Provisioning of External Systems .....	48
<b>3. Dual-Version PortaSwitch® .....</b>	<b>50</b>
Concept of Dual-Version PortaSwitch® .....	51

## Preface

This document provides PortaSwitch users with the description on the system architecture, principles of operation and provides examples for the installation and maintenance.

### Where to get the latest version of this guide

The hard copy of this guide is updated upon major releases only, and does not always contain the latest material on enhancements that occur in-between minor releases. The online copy of this guide is always up to date, and integrates the latest changes to the product. You can access the latest copy of this guide at [www.portaone.com/support/documentation/](http://www.portaone.com/support/documentation/).

## Conventions

This publication uses the following conventions:

- Commands and keywords are given in **boldface**.
- Terminal sessions, console screens, or system file names are displayed in `fixed width font`.



**Exclamation mark** draws your attention to important actions that must be taken for proper configuration.

**NOTE:** Notes contain additional information to supplement or accentuate important points in the text.



**Timesaver** means that you can save time by taking the action described here.



**Tips** provide information that might help you solve a problem.



**Gear** points out that this feature must be enabled on the Configuration server.

## Trademarks and Copyrights

PortaBilling®, PortaSIP® and PortaSwitch® are registered trademarks of PortaOne, Inc.

# Hardware and Software Requirements

## Server System Recommendations

Five (5) UNIX Servers for the PortaBilling® or ten (10) servers for PortaBilling® Procinctus. For additional details regarding the recommended hardware configuration of each server, consult the [Hardware Recommendations](#) section on our website.

For information about whether particular hardware is supported by Oracle Enterprise Linux used as the operating system in PortaSwitch®, consult the related document on the Oracle or RedHat website: <https://hardware.redhat.com/>.

## Client System Recommendations

- **OS:** MS Windows XP or above, Linux/BSD, Mac OS X 10.6 or above.
- **Web browser:**
  - Google Chrome 55 or above, Mozilla Firefox 50 or above.
  - JavaScript and cookies must be enabled.
- **Spreadsheet processor:** MS Excel, OpenOffice Calc, LibreOffice Calc, Google Sheets.
- **Display settings:** A minimum screen resolution of 1024 × 768.

# Disk Space Requirements

When buying hardware for your servers, you obviously want to be sure that their capability will be sufficient to suit the demands of your business. Among other hardware requirements, the HDD capacity should be carefully considered to ensure that it will satisfy the desired traffic patterns as well as future business growth opportunities.

Disk space requirements can be estimated based on business scenarios, traffic patterns and the desired PortaSwitch® configuration. Consider the examples below.

## PortaBilling®, PortaSIP® and log storage servers

We will use the following figures to estimate the disk space required:

- Each call processed by PortaSwitch® will increase SIP logs by approximately 100 KB and billing logs by approximately 50 KB.

- Each registration processed by PortaSwitch® will increase SIP logs by approximately 10 KB and billing logs by approximately 5 KB.

We will also use the following assumptions for the sake of simplicity:

- The average call duration is around 5 minutes.
- A call success rate (ASR) is 50% (the industry norm).
- There are ten thousand registered phones with an average re-registration period of 5 minutes.
- The PortaSIP® server and the PortaBilling® servers contain uncompressed log files for the last three days and all log files for the required period (e.g. 10 days) will be available on the centralized log storage (7 compressed and 3 uncompressed).
- Log compression ratio is 5% (bzip2).

Using the figures and assumptions described above, we can estimate how much disk space is needed to process 10 MMM (ten million minutes per month). Taking into consideration that for every failed call, logs and xDRs are also generated, we need to multiply the total sum by two (ASR 50%). Note that we calculate an average number of calls per day using the following formula:  $10,000,000 \text{ minutes} / 5 \text{ minutes} / 30 \text{ days} = 66,666$  calls and then round up this number to 70,000, for convenience.

- *PortaBilling®:*
  - Calls:  $70,000 \text{ calls per day} * 50 \text{ KB} * 2 \approx 7 \text{ GB}$  for one day
  - Registrations:  $288 \text{ registrations / day per phone} * 10,000 \text{ phones} * 5 \text{ KB} \approx 14.4 \text{ GB}$  for one day
  - Total for one day:  $7 \text{ GB} + 14.4 \text{ GB} = 21.4 \text{ GB}$
  - Total for three days:  $21.4 \text{ GB} * 3 \text{ days} \approx 64.2 \text{ GB}$
- *PortaSIP®:*
  - Calls:  $70,000 \text{ calls per day} * 100 \text{ KB} * 2 \approx 14 \text{ GB}$  for one day
  - Registrations:  $288 \text{ registrations / day per phone} * 10,000 \text{ phones} * 10 \text{ KB} \approx 28.8 \text{ GB}$  for one day
  - Total for one day:  $14 \text{ GB} + 28.8 \text{ GB} \approx 42.8 \text{ GB}$
  - Total for three days:  $\approx 42.8 * 3 \text{ days} \approx 128.4 \text{ GB}$
- *The centralized log storage:*

$$3 \text{ days} * (21.4 \text{ GB} + 42.8 \text{ GB}) + 7 \text{ days} * (21.4 \text{ GB} + 42.8 \text{ GB}) * 5\% \approx 192.6 \text{ GB} + 22.5 \text{ GB} \approx 215 \text{ GB}$$

## Database servers

We will use the following figures to estimate how much disk space is needed on the database servers:

- Each xDR takes up about 1.5 KB database space.

- PortaBilling® produces at least 2 xDRs for each call. Actually, the number of xDRs produced depends on the call scenario and can reach up to 5–7 or even more xDRs for complex calls. For simplicity's sake, we assume that on average, 3 xDRs are produced per call.

We will also use the following assumptions for the sake of simplicity:

- The average call duration is around 5 minutes.
- A call success rate (ASR) is 50% (the industry norm).
- You will need to allocate space for MySQL binary log files (25–100 GB depending on the rate of usage of the database). Let's allocate 50 GB for binary log files.
- In addition, for performing operations such as backup, you will need to reserve an amount of free space roughly equal to the projected database size.
- You will keep xDRs for the previous 60 days.

Using the figures and assumptions described above we can estimate the disk space that will be consumed to process 10 MMM (ten million minutes per month) on the database servers:

- $70,000 \text{ calls per day} * 1.5 \text{ KB} * 3 \text{ CDRs} * 60 \text{ days} * 2 * 2 \approx 75.6 \text{ GB}$

So, not taking into the consideration the MySQL binary log files and reserving space for operations such as backup, 75.6 GB is required on the database servers to process 10 MMM.

### Estimates based on different traffic patterns

The following table shows the minimum amount of disk space required on each server for installations with different traffic patterns. This includes the 60 GB minimum necessary for system installation.

Traffic Pattern	The Database Servers	PortaBilling Server	PortaSIP server	The Centralized Log Storage Server
		<i>keep logs for last two days only</i>		
10 MMM, 10,000 registered phones	185 GB	124 GB	188 GB	275 GB
20 MMM, 20,000 registered phones	260 GB	188 GB	316 GB	490 GB

50 MMM, 50,000 registered phones	485 GB	220 GB*	273 GB**	1 TB
100 MMM, 100,000 registered phones	860 GB	380 GB*	487 GB**	2.2 TB

*\* If two PortaBilling® servers provide AAA services, then this amount of disk space will be needed on each server.*

*\*\* If three PortaSIP® servers process traffic, then this amount of disk space will be needed on each server.*

**NOTE:** The above figures are minimal and depending on the services provided, may grow (e.g. call records take up disk space. In order to store 15 hours of recorded conversations, 1 GB of disk space is required).

## Installation

PortaSwitch® installation ISO files contain everything required for installing Oracle Enterprise Linux (64-bit version), PortaSwitch® and the supplementary packages that are necessary for convenient system administration and maintenance.

After the installation is complete you will add PortaSwitch® applications (e.g. RADIUS server, web server, etc.) to individual servers using the Configuration server tool – this will automatically enable the required components of PortaBilling®Switch® software on each server.

This allows you to install a completely functional PortaSwitch® environment (multiple servers) from scratch in less than one hour!

For detailed installation instructions, the recommended network configuration and the optimum setup for PortaSwitch® behind a firewall, please refer to the [PortaBilling® Installation Guide](#).



## What is New in Maintenance Release 65

### Added:

- The [Call Delivery in Dual-Version PortaSwitch®](#) section to the Dual-Version PortaSwitch® chapter.
- The [Common API Entry Point for Dual-Version PortaSwitch®](#) section to the Dual-Version PortaSwitch® chapter.

### Updated:

- The [Per-configuration Licensing: Flexibility and Control](#) section.

# 1 ■ System Architecture

## Overview

PortaSwitch® is a unified platform for telecommunication service providers, wholesale carriers, ISP, MVNO and NGN operators for unifying voice and data traffic within a single converged network. It provides various prepaid, postpaid, retail and wholesale telecommunication services, including calling cards, Vonage- and Skype-like services, CLEC type services, MVNO & MVNE, ISP, WiMAX & WiFi and much more.

The main components of PortaSwitch® are:

- PortaBilling® – Real-time converged billing and service provisioning system.
- PortaSIP® – A class 4 (SBC) and class 5 SIP softswitch with media application (PortaSIP® Media Server) that plays IVR (voice prompts).

On the network level, PortaSwitch® communicates with IP phones, communication clients (on PCs, smartphones, etc.) and VoIP gateways via the SIP protocol. There are no restrictions as far as the vendor or model of the equipment – basically any communication device which supports SIP can be used in conjunction with PortaSwitch® for services such as voice or video calls, presence or instant messaging.

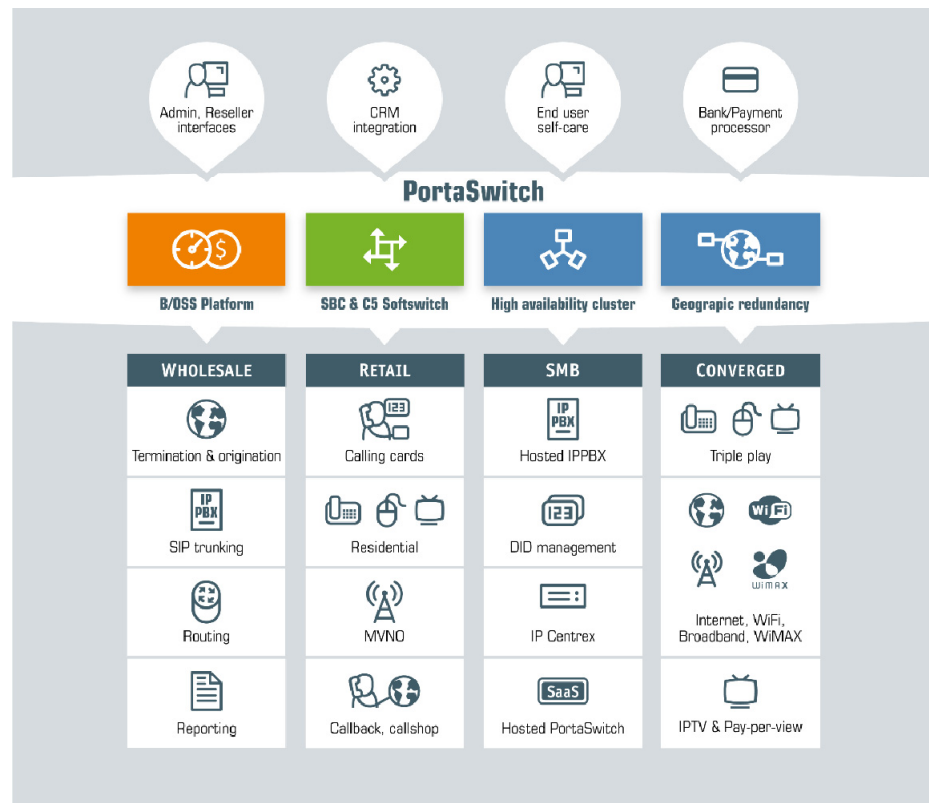
The PortaBilling® component stores all the information about your products, rates and customers and the service configuration for individual customers or phone lines. It is managed via a web interface and includes a self-care portal for your end users. PortaBilling® controls call processing on PortaSIP® (whether a caller is allowed to make a call) as well as real-time routing (which carriers and in which order should be used to send a call in order to maximize profit and provide the required quality of service).

PortaBilling® is a converged system; it can be used as a single administration interface to manage (or bill for) multiple services, including those provided by third-party network elements (for instance, LTE SAE-GW or WiMAX ASN-GW), while charges for the different services will be grouped on a single bill.

The key difference between PortaSwitch® and more traditional “switch” products is that PortaSwitch® offers much more. It includes the B/OSS component, and so is a unified service management and delivery platform.

As a service provider, you want not only to let customers make phone calls, but also to use the platform as your main source of revenue generation. Thus PortaSwitch® provides real-time verification of available funds, detects and prevents potential fraudulent activity, automatically

disconnects calls to prevent balance overdrafts, provides flexible rating of service usage, offers a tool to create attractive product bundles, automatically assesses monthly recurring charges, generates and delivers invoices electronically, and enforces the payment collection process.



## Centralized Configuration Management

In order to efficiently maintain large PortaSwitch® installations (which may involve 10 or more servers), it is essential to have a unified interface for managing all the configuration data. Tasks such as IP address changes, relocating services to different physical servers, or simply changing an option that affects functionality can then be performed quickly and easily, with a minimal chance of error.

Configuration server carries out exactly this task, providing an interface for the administrator to view the current configuration, create a new configuration and correctly apply it to all servers, or rollback to an old configuration if a problem has been detected. Another important role of the Configuration server is that it stores “images” of different versions of the software. Each image is the actual content (in a binary format) of a specific version of the software code (e.g. Maintenance Release 55, build 1). When a specific image is loaded, the server will operate under the corresponding software release.

## Concepts

There are several important concepts involved in the configuration management framework. Configuration management is designed to work in the same way whether it is controlling just a single PortaBilling® installation (four servers) or a PortaSwitch® Procinctus (ten servers). In the rest of this section, we will use some examples related to managing the full PortaSwitch® configuration (five servers), so as to better illustrate the capabilities of the configuration framework.

### Server

A server is an atomic element of processing capacity. It is either a single physical server, or a separate virtual machine, if virtualization is used. In other words, it is basically a host on which PortaSwitch® software can be installed and operated. A server has attributes such as a number of available CPUs, disk space, and so on.

### Private Cloud

Several servers within the same PortaSwitch® installation make up a private cloud environment. They all run the same version of the software and, apart from differences in the available hardware resources (e.g. one server has a faster CPU), they are completely interchangeable – i.e. a PortaSwitch® software component that can run on server A can also run on server B.

### Instance

An instance is a copy of an application (e.g. RADIUS server) configured in a particular way and running on a server, i.e. it is a combination of the software code, configuration data, and running processes that provides an actual service. For example, a PortaBilling® RADIUS instance with IP address 1.2.3.4 may be created on server ABC. Or three instances of PortaSIP® processing node may be created on server XYZ. They all have different IP addresses, and may differ in other configuration parameters, e.g. one of them has the “start accounting” option turned on while the other two have it turned off.

### Application Server

An application server is a server with one or more running application instances of a certain type. For example, three instances of PortaSIP® processing nodes are running on server ABC and one instance of PortaSIP® processing node and one instance of the billing engine are running on server XYZ. In this case you have two PortaSIP® application servers and one billing engine application server.

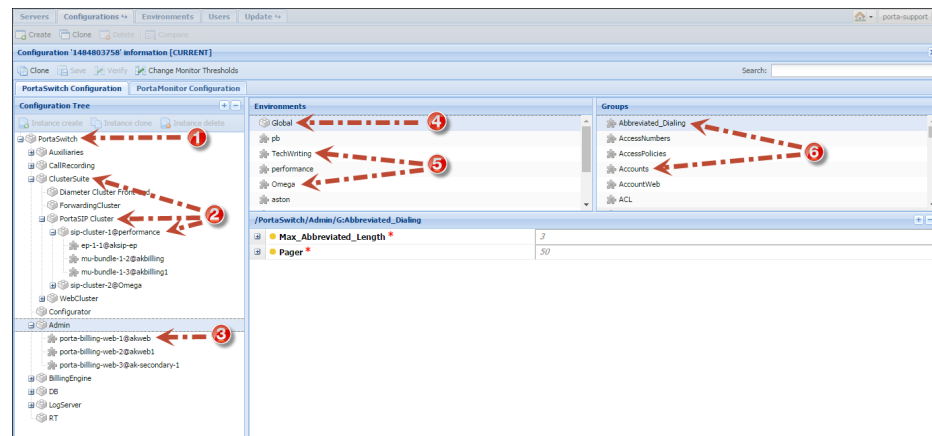
## Option

An option is a configuration parameter which alters the system's functionality. Some examples of options would be “When should statistics generation be done,” or “Should the previous balance be included in the invoice's amount due.” Depending on an option, you can set its value by selecting it from a list, or by typing it into a text field.

For convenience in administration, a default value is provided for most of the options, so that you do not have to supply a value for every single option in order to make the system work.

## Configuration tree

The full system configuration includes hundreds of different options, so it would certainly be inconvenient to work with them as a single list. Thus options are grouped together in a tree-like structure.



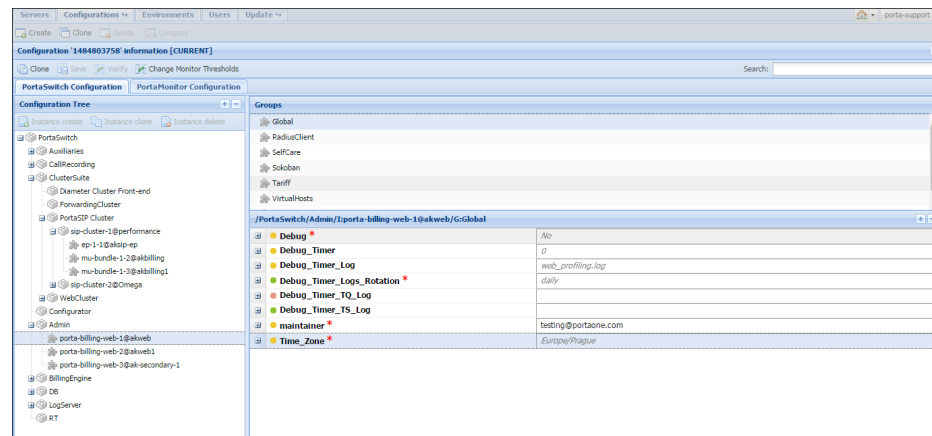
There are global options at the top level of the tree (**PortaSwitch** (1)), i.e. those that have an effect on all of the components of the system.

Beneath the global level is the application level (2). Each application is presented as a separate node in the configuration tree. These nodes can be grouped under bigger logical nodes, i.e. sip-cluster nodes are grouped under the **PortaSIP Cluster** node, which is itself is a subnode of **ClusterSuite**.

Thus, in order to change the **allow\_reauth** option (which is related to the PortaBilling® application), you would go to the desired sip-cluster node of the PortaBilling® Cluster: **PortaSwitch**→**Cluster Suite**→**PortaSIP Cluster**→**sip-cluster@193.168.1.17**→**MUB2bua**.

Finally, there is the instance level (3), which covers options related to a specific instance.

For example, in order to change the **Time\_Zone** option for the porta-billing-web instance with IP address 193.28.87.21, you would go **PortaSwitch→Admin→porta-billing-web-1@193.168.1.17**.



Some options may have different values in different virtual environments. These options are organized into environment sets, and each set provides control for options that are specific to a particular virtual environment. Virtual environments have their own hierarchy; the **Global** set (4) is the highest level and individual virtual environments (5) represent the lower level.

Since there are normally many individual options available at each level, for management's convenience, the options are split into groups (6), and each group contains a small set of options that is related to a single software module or feature.

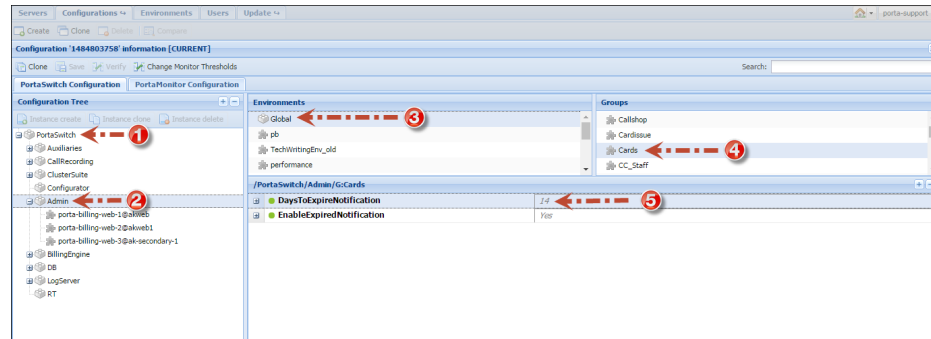
When the value for an option is defined at a certain level, it is automatically propagated to all of the levels beneath it – if no value is directly specified for this option at a lower level.

This means that the Configuration server chooses which value to use according to the following rule:

- If you specify an option value at a lower level, the Configuration server uses this value and ignores the data specified at a higher level;
- If you leave the option value at a lower level undefined, the Configuration server uses the *default* value – that is, the one specified at a higher level.

Consider the following example.

In the **Configuration Tree** panel you can go to the **PortaSwitch→Admin** node and in the **Global** set specify 14 as the default value for the **DaysToExpireNotification** option (you can find this option in the **Cards** option group).



This value then becomes the default value for all of the virtual environments (those that exist and those not yet created), and PortaBilling® will send notifications to customers about their card expiration 14 days before the expiration date.

However, if you need to notify customers in one of your billing environments (e.g. pb) 10 days before their cards expire, set the **DaysToExpireNotification** option value to 10, exclusively for this environment.

## Multiple configurations

When a configuration is saved, this stores all the options in it – so every stored configuration contains a complete set of data for operating all PortaSwitch® components. In order to preserve system integrity it is not possible to directly alter the active configuration (the one currently applied to the servers).

In the case of changes not producing the desired result, it is always possible to roll the system back to its original, stable state.

The process of changing the system configuration is thus divided into several steps:

1. Clone the current configuration tree into a new one.
2. Make the required changes.
3. Now apply this configuration to the system so that it becomes the **active** one.

## Applying the configuration

Every server in PortaSwitch® runs a configuration update agent, which follows commands from the Configuration server. When a configuration update is received, the agent updates the local files, restarts the processes and does everything else required to put the changes into effect.



## Per-configuration Licensing: Flexibility and Control

The license provides you with great flexibility when deploying the PortaSwitch® installation on servers. It doesn't constrain your business parameters (e.g. the number of concurrent calls, ports, minutes or end users) and it defines the type of PortaSwitch® installation you have. The license is applied per PortaSwitch® installation (site) and is defined by the following parameters:

- The *maximum* number of **servers** (where you add *any* PortaSwitch® applications) in your installation;
- The *maximum* number of specific **application servers** in your installation. In particular:
  - The *maximum* number of servers where you can deploy a billing engine (RADIUS / Diameter).
  - The *maximum* number of servers where you can deploy PortaSIP® (PortaSIP® dispatching node or PortaSIP® processing node).
  - For those products that have a bundled license for Oracle software – the *maximum* number of servers where you can deploy Oracle database cluster software may *not* exceed the number of billing engine servers (e.g. if your license includes five billing engine servers, then you can deploy a *maximum* of five servers with Oracle database cluster software).

Consider the following example:

A standard PortaSwitch® Procinctus installation license covers ten servers. Among them, two servers run the billing engine cluster, two servers run the web cluster and three servers run the PortaSIP® cluster.

An ITSP decides to deploy PortaSwitch® Procinctus on only eight of their servers:

Nº	Server Name	Running PortaSwitch Applications
1	Alpha	<ul style="list-style-type: none"> <li>• Billing engine</li> </ul>
2	Beta	<ul style="list-style-type: none"> <li>• Billing engine</li> <li>• Master DB</li> </ul>
3	Gamma	<ul style="list-style-type: none"> <li>• Web server</li> </ul>
4	Epsilon	<ul style="list-style-type: none"> <li>• Web server</li> <li>• Replica DB</li> </ul>
5	Zeta	<ul style="list-style-type: none"> <li>• PortaSIP® dispatching node</li> </ul>
6	Iota	<ul style="list-style-type: none"> <li>• PortaSIP® dispatching node</li> <li>• PortaSIP® processing node</li> </ul>

7	Sigma	<ul style="list-style-type: none"> <li>• PortaSIP® processing node</li> </ul>
8	Omega	<ul style="list-style-type: none"> <li>• Configuration server</li> </ul>

This configuration will work just perfectly; furthermore, it is possible to easily add one more server and configure additional PortaSIP® processing node instances there for enhanced productivity.

As you see, you can combine different PortaSwitch® applications (e.g. web server and PortaSIP® server instances) on a single server and easily add application instances to your servers to scale up your capacity and match your unique business requirements.

An exception is the Configuration server instance, which, due to its specific purpose, is only compatible with the Log server instance.

At the same time, the total number of servers in your installation and number of application servers with the billing engine and PortaSIP® instances must not exceed the parameters defined by your license.

### What's in it for me?

A license verification method is based on centrally distributed **license files** and provides the user with a flexible and convenient service management system. As PortaSwitch® applications (e.g. PortaBilling RADIUS or PortaSIP) are not bound to a specific physical server, you can change the system configuration and launch applications on new servers without any need for your physical presence in the facility where your servers are located.

PortaSwitch applications can be moved between servers with ease – you can turn a server that used to be a web server into a PortaSIP® or add any other applications with just a few mouse clicks on the web interface of the Configuration server.

The time required to deploy new applications after the license has been purchased is minimum. For example, if a rapid increase in traffic is anticipated this coming weekend, you can contact the PortaOne sales team and once your purchase has been finalized and the license information has been updated in our CRM, you can immediately add extra PortaSIP® instances to your system in just a few minutes.

A hardware failure no longer causes a lengthy service outage. If, for example, the RADIUS server you were running goes down because of a hardware failure – you can promptly move the service to a different host. This eliminates several hours of potential downtime, since there is no need for someone to travel to the collocation facility where the servers are installed. While the RADIUS is running on a different server, you have plenty of time to fix (or even replace) the defective one. By the same

token, you can add new physical servers or perform maintenance on existing ones without interrupting the flow of your business, by reassigning the applications to other servers.

### What is a License file?

It is a protected .xml file that contains the following information:

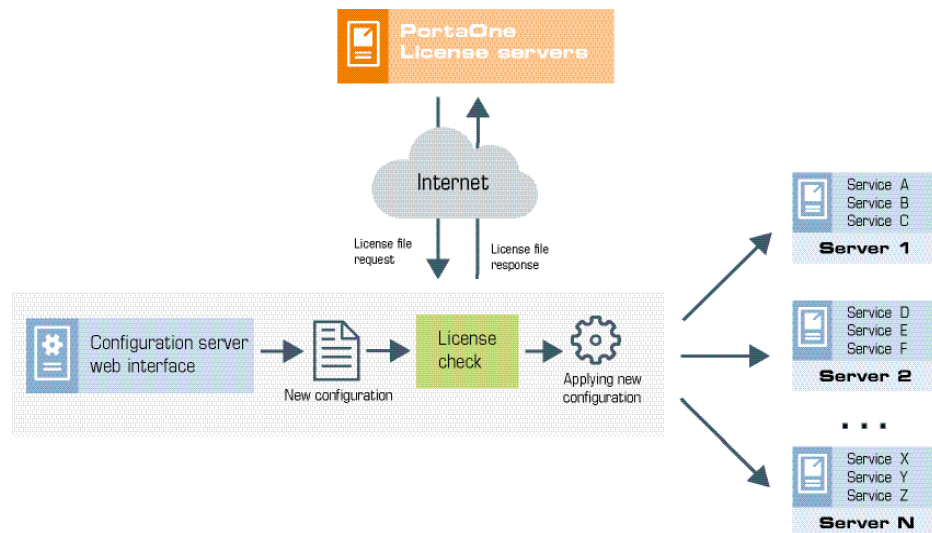
- Component instances (e.g. PortaSIP®, billing engine, DB, web, RT, etc.).
- Instance options (e.g. cluster, SMP etc.).
- Information about IPs.
- Expiration date.
- Information about the owner.
- Encryption seed and signature.

In general, it is similar to an email signed with a PGP and looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<License>
  <Instance Server_IP="10.17.190.3" Node="OracleDB" Service_IP="10.17.190.17">
  </Instance>
  <Instance Server_IP="10.17.190.35" Node="PortaBE" Service_IP="10.17.180.249">
    <Option Name="Radius SMP">Yes</Option>
    <Option Name="Radius Cluster">Yes</Option>
    <Option Name="Minutes per month">0</Option>
    <Option Name="Radius Engine Count">64</Option>
  </Instance>
  <Instance Server_IP="10.17.190.253" Node="PortaSIP" Service_IP="10.17.180.201">
    <Option Name="Number of sipenvs">20</Option>
  </Instance>
  <Instance Server_IP="10.17.190.253" Node="PortaPresence" Service_IP="10.17.180.173">
    <Option Name="Number of presence envs">20</Option>
  </Instance>
  <Instance Server_IP="10.17.190.251" Node="PUMServices" Service_IP="10.17.180.251">
  </Instance>
  <Instance Server_IP="10.17.190.251" Node="PUMPeriodicTasks" Service_IP="10.17.180.251">
    <Option Name="Number of mp3 encoding threads">5</Option>
  </Instance>
  <Instance Server_IP="10.17.190.251" Node="VoiceMailDB" Service_IP="10.17.180.251">
  </Instance>
  <Instance Server_IP="10.17.190.34" Node="PortaBE" Service_IP="10.17.180.250">
    <Option Name="Radius SMP">Yes</Option>
    <Option Name="Radius Cluster">Yes</Option>
    <Option Name="Minutes per month">0</Option>
    <Option Name="Radius Engine Count">64</Option>
  </Instance>
</License>
```

### How does it work?

You can make changes to your “live” system at any time (e.g. add a new PortaBilling® RADIUS instance or move it to a different physical server) using the Configuration server web interface (see Section 2 of the [PortaSwitch Configuration Server Web Reference Guide](#)). When you apply the change, the Configuration server will retrieve the **license file** from a centralized PortaOne Licensing Server and check whether all of the new configuration items (e.g. total number of RADIUS servers in the cluster) are in line with the license terms. If the configuration corresponds to your license, it will be applied; otherwise you will be prompted to change the configuration so that it meets license restrictions.



A local copy of the license file is stored on the Configuration server and then distributed to the remaining servers. Each individual application uses it to verify that this service can run as a part of this installation – a valid license file is necessary for any application to operate. The local copy of the license file is updated every night to prevent it from expiration.

**NOTE:** For your installation to work properly, PortaOne Licensing Servers ([license1.portaone.com](http://license1.portaone.com), [license2.portaone.com](http://license2.portaone.com)) should be accessible from all your hosts.

When none of the licensing servers are accessible, the monitoring system shows a corresponding warning message. To make sure your business is not affected by a problem with Internet connectivity, preventing your servers from contacting PortaOne Licensing Servers, the license file will be valid for a week after download. That is, even in the unlikely event that for several consecutive days your server does not have connectivity to the Internet and cannot access any of the licensing servers, your services will continue running for up to seven days, which is quite enough time to restore access.

## Deploying PortaSwitch® Across Multiple Sites

To meet customers' expectations regarding the quality of communication services the service provider needs to introduce an extra degree of reliability within the network and its applications, so that the service is not interrupted – even if some network components are not functioning. How can this demand be addressed?

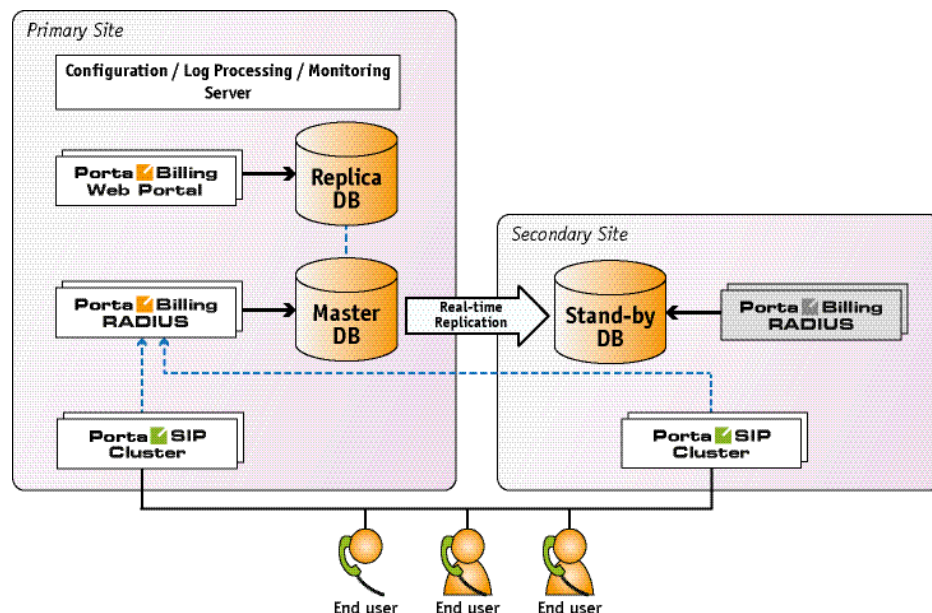
The per-server redundancy (when there are two physical servers and each runs a copy of an application, such as PortaSIP®) addresses the situation

when a single server fails (e.g. hardware fault). But there is another class of “catastrophic” events that can render all servers installed in the same location (rack, hosting center, etc.) unavailable. Such events include natural disasters, power outages at the collocation provider, network routing errors, etc. The only way to overcome this and provide uninterrupted service is to have another set of servers in a different location that can continue operating during the outage at the “main” site.

It is important in this situation that the “secondary” site not only activates and begins providing service as soon as possible, but also that it automatically synchronizes the changes later on (updates balances, xDRs, etc.) to the “main” site.

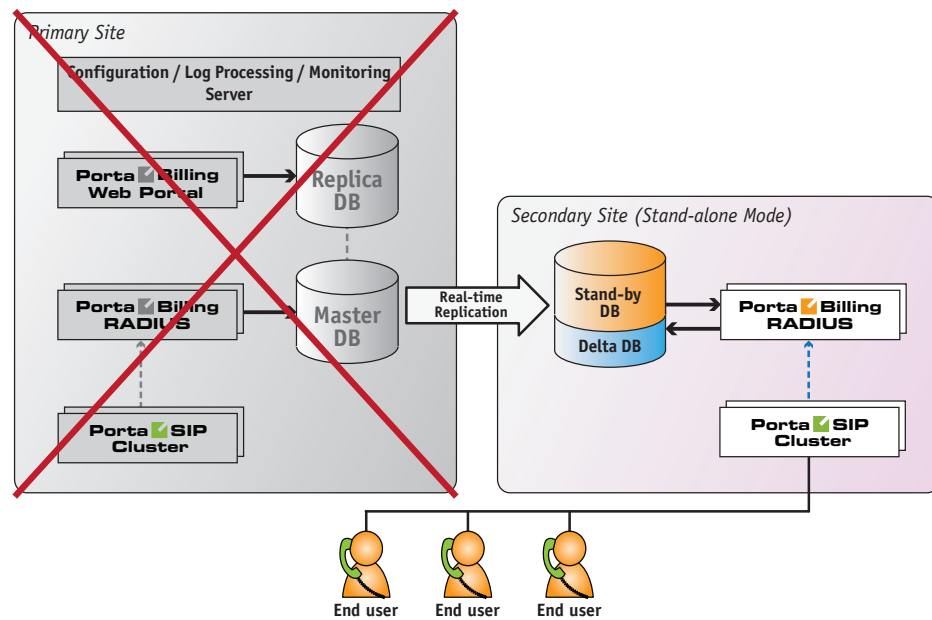
All of the above is available as the PortaSwitch® **site redundancy** solution, which allows service providers to:

- Protect themselves against hosting facility outages.
- Provide service to multiple geographic regions – even if network connectivity between those regions is lost.
- And finally, perform upgrades to new software versions with zero downtime! This last provision adds an essential benefit to the deployment of PortaSwitch® across multiple sites, since although one might hope that a hosting facility outage would never happen, one can be certain that sooner or later, there will be a need to perform a software upgrade.

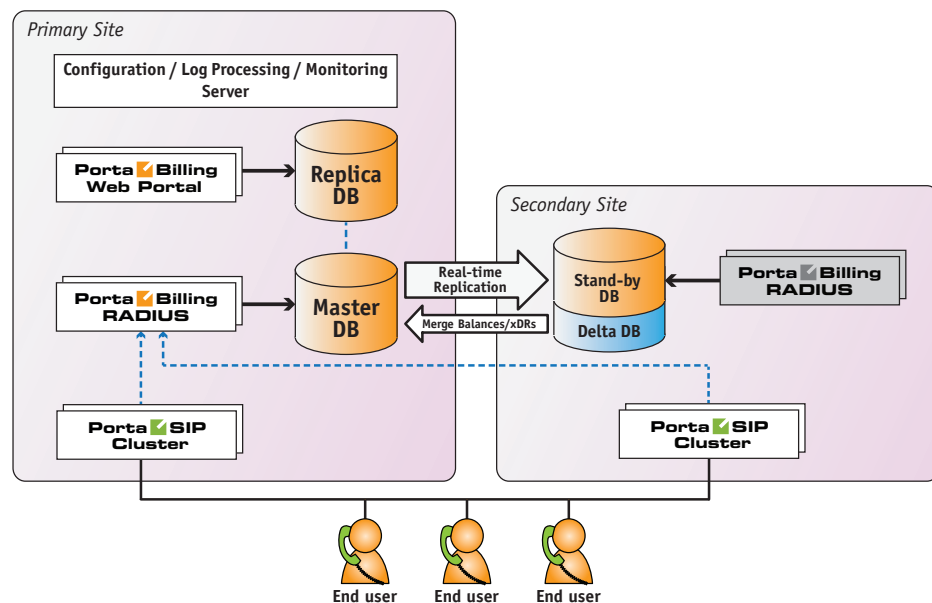


So if the secondary site detects that the main site has become unavailable, the “stand-alone” mode is activated on the secondary site and now it provides the service to end users using the latest available snapshot of the service configuration. The xDRs for consumed services and changes in balance are accumulated in a separate database (on the stand-by database

server) and are taken into consideration when authorizing subsequent activities, so there is no risk of balance overdraft when the stand-alone mode is used.

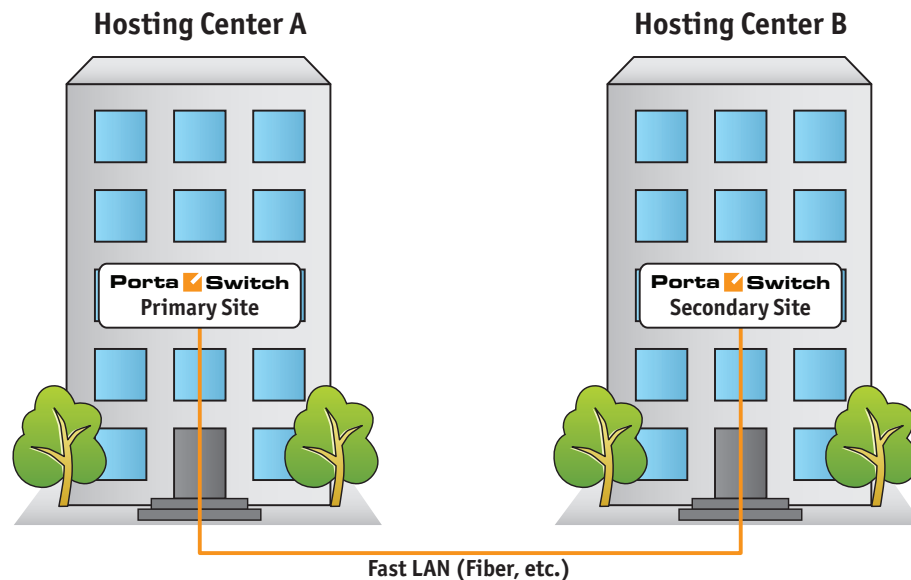


Once the main site becomes available again, the secondary site starts the process of synchronizing all of the accumulated changes to the main site and then the secondary site switches back to its normal (“stand-by”) mode.



## Typical Deployment Scenario

Let's consider the example of a possible PortaSwitch® deployment across multiple sites. The “primary” site hosts a standard PortaSwitch® Procinctus (the Configuration server, main and replica database servers, a cluster of PortaBilling® RADIUS and web servers and the PortaSIP® cluster). The “secondary” site is located in a different hosting facility (with a fast connection to the main site) and contains the stand-by database server, PortaBilling® RADIUS server and the PortaSIP® cluster.

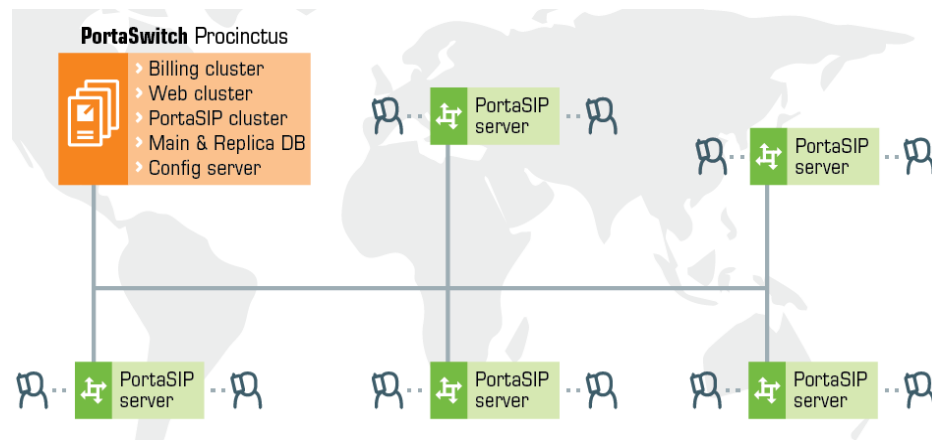


Within its “normal” mode of operation at the remote site:

- The stand-by database server continually retrieves changes from the main site, so it always has an up-to-date snapshot of the database from the main site.
- The RADIUS billing servers are in “stand-by” mode, so they do not actively process any requests.
- The PortaSIP® cluster provides service as usual (processing incoming calls, playing the IVR, etc.). It uses the RADIUS servers on the **main** site for authentication and writes any changes (e.g. updated SIP phone location) into the primary database.

Another option is deploying secondary site (or sites) in a different city or country using WAN connectivity.





## When Disaster Strikes

If there is an outage (for instance, a motherboard failure) on a single server (e.g. PortaBilling® RADIUS server #1) at the primary site, the primary site continues to operate as usual. Another server within the cluster (PortaBilling® RADIUS server #2 in our example) processes all the requests and there is no need to switch over to the secondary site.

The above statement is true for an outage on any server **except** the primary database, since an outage there would render all other servers on the primary site (billing engine, PortaSIP®) unable to function normally.

Therefore, the activation of the stand-alone mode on the secondary site would only happen if:

- There is an outage on the primary database server.
- There is an outage on all servers at the primary site (e.g. power failure).
- There is a network outage that makes the primary site inaccessible from the secondary site.

In this case, the **stand-alone mode** would be activated on the secondary site. This is a special mode of operation that allows the site to provide as many services (e.g. placing outgoing calls, receiving incoming calls, accessing IVR auto attendant, placing calls using calling card IVR, etc.) for end users as is still possible. At the same time, we assume that the outage at the main site is (most likely) temporary, so when order is restored, synchronization with the primary site will need to be performed. In stand-alone mode, certain operations are disabled if they could cause a breach in data integrity between the sites – for instance, it would not be possible to create new accounts, change service configurations, etc.

When a service is provided on the secondary site, the billing engine continues to calculate applicable charges according to product, tariff and the responsible party's other billing parameters (e.g. from the account that



originated the call). Changes to the balance and new xDRs are written into a separate database (the “delta” database, which runs on the same physical server as the stand-by database). This allows the billing engine to keep track of already consumed services and avoid a balance overdraft – even if a secondary site has to operate in stand-alone mode for an extended period of time – and this, therefore, results in a clear history of all produced charges. When the primary site becomes available again, these changes are automatically applied to the primary database – and the secondary site is switched back to “normal” mode. All of this happens automatically, without any need for PortaSwitch® administrator involvement – and an end user might not even notice that there were any problems at the main site.

## Example Scenario

Let’s detail what happens in case of a primary site outage using a single customer as an example. The customer “ABC” has account number 12345 provisioned on his IP phone. The customer has a current balance of \$98.00, a credit limit of \$100 and his rate for calls within the US is \$0.10/minute. The primary and secondary sites are configured as previously described.

- A power outage makes the entire primary site unavailable.
- This event is detected by a watchdog script on the secondary site so it switches into “stand-alone” mode (in particular, this enables the RADIUS server on the secondary site and instructs the PortaSIP® cluster on the secondary site to use it as the authorization source).
- If the user’s SIP phone was previously registered to the PortaSIP® cluster on the primary site, during the next re-registration attempt the phone will detect that the cluster is no longer available and attempt to contact an alternative server (this list is either pre-programmed into the phone or obtained dynamically using DNS). When it reaches the PortaSIP® cluster on the secondary site it registers there. (If the phone is already registered on the PortaSIP® cluster on the secondary site, nothing changes.)
- When the user attempts to make an outgoing call, an authorization request is sent to the PortaBilling® RADIUS server on the secondary site.
- The billing engine uses the currently available balance information (\$98.00) to compare it with the credit limit (\$100.00) and authorizes the call for no more than 20 minutes.
- When, after 12 minutes of conversation, the user hangs up, PortaSIP® sends an accounting request to PortaBilling® so that charges are applied.
- When PortaBilling® processes the request, it calculates the amount to be charged (\$1.20) and stores the balance adjustment

(\$1.20) and the xDR for that call (with all call details such as CLI, CLD, call connect time, etc.) in the delta database.

- Then, when the user makes another call and PortaSIP® sends an authorization request, the billing engine calculates the “effective” balance as the sum of the balance in the stand-by database (\$98.00) and the balance adjustment stored in the delta database (\$1.20). So the effective balance is \$99.20 and the call will have a time limit of 8 minutes.
- The user hangs up after 5 minutes, so there is another xDR for that call with the charged amount of \$0.50 written to the delta database and the balance adjustment is now \$1.70.
- The next call will only be authorized for the remaining \$0.30 of available funds – and can only run until the balance reaches the credit limit. This prevents balance overdraft – even if the site operates in stand-alone mode and the balances in the stand-by database are not changed.
- When the primary site comes back up, synchronization takes place.
- First to happen is that funds in the amount of the balance adjustment (\$1.70) are locked in the primary database – this ensures that if a customer now tries to use the service on the main site, he will only be able to spend the \$0.30 that he has available.
- Next, the secondary site is switched back to “normal” mode.
- And then, individual xDRs are transferred to the primary database.

This two-step process (first funds lock, then actual xDR transfer) ensures the avoidance of balance overdraft on the main site while an xDR transfer is in progress. There can be a large number of xDRs (if a secondary site operated in stand-alone for an extended period of time) and consequently, it can take time to replicate all of them to the primary site.

## Stand-Alone Mode Restrictions

The secondary site does not differentiate between these two types of events:

- The primary site is down or has been destroyed (power failure, hurricane, earthquake, etc.).
- The primary site is still up and operational, but connectivity between the primary site and the secondary site is lost. For instance, the primary site is in city A and the secondary site is in city B. So while there is no connectivity between those two city sites, each one functions normally; in each city there are users using the service.

When the secondary site operates in stand-alone mode, it is essential that data integrity between the primary and secondary sites is protected at all times. This means that no operations should be allowed to run on the secondary site that could cause data conflict when merging data change back to the primary site.

Let's assume that during a connectivity outage between the sites the service configuration is changed as follows:

- The end user, connected to the secondary site, sets up call forwarding to phone number 123.
- On the primary site the administrator also sets up call forwarding for this user to phone number 456.

Once connectivity between the sites is restored and a data merge is performed, it could be unclear which configuration could be regarded as valid (i.e. which number would end up as the forwarding number). This is called a “split brain” problem and, of course, must be prevented from happening.

So although the secondary site can detect that the primary site is not accessible, it regards the primary site as operating normally, since users are making calls, administrators are making changes to the web interface and data is being changed there. Thus, the secondary site (when activated) does not perform all of the functions of the primary site; stand-alone mode requires that some functionality must be disabled.

In short, in stand-alone mode, the only operations allowed are those that change the balance and produce xDRs. All other changes (e.g. changing service configuration attributes or creating new entities) are prohibited. While the secondary site is in stand-alone mode, users can make and receive all kinds of phone calls (using IP phones or calling card IVRs) including such complex scenarios as call pickup, call transfer, etc. They can also use IVR applications that do not change the service or account configuration. Such IVR applications are fully available in stand-alone mode and are as follows:

- Account top-up via voucher.
- Email callback.
- Balance information.
- One-stage calling.
- Pass-Through IVR.
- WEB callback.
- Conferencing.

Some of these IVR application components / commands modify the service or account configuration; therefore, they are available with limitations in stand-alone mode:

- Callback calling (account registration is disabled).

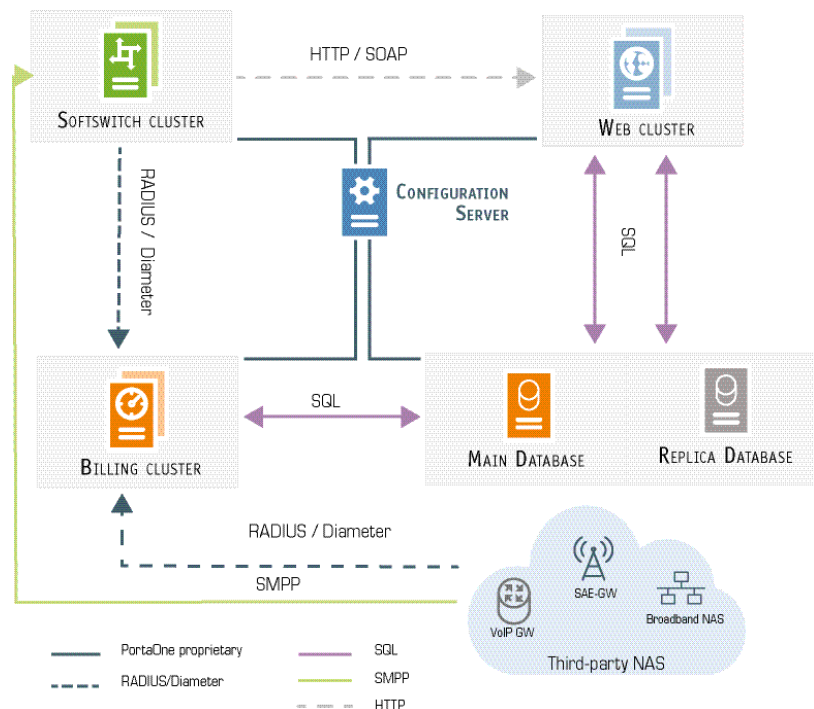
- Screening IVR (fraud protection is disabled).
- Prepaid card calling (account registration is disabled).
- SMS Callback (account registration and change password commands are disabled).
- Voicemail (messages are placed into the exim mail queue).
- Call Queues in auto attendant (the first caller in the queue may become disconnected before the secondary site is switched to the stand-alone mode).

The following IVR applications change the service or account configuration and are **not available** in stand-alone mode:

- Account self-care.
- Account top-up via credit card.
- Call forwarding management.
- Access to one's own voice mailbox.
- Payment Remittance – Transfer To.

Access to the web interface is also disabled.

## PortaSwitch® Application Components



Every PortaSwitch® application (e.g. RADIUS server, web interface, etc.) consists of a combination of processes (subcomponents such as the RADIUS daemon, the Apache server, statistics collection tools, etc.). All of these processes possess certain functions and interact with a number of other processes. If one process fails, a service that is using it may stop

working (even if this service is provided from another server). Hence, it's important to understand the interconnection among PortaSwitch® applications (and their subcomponents) and learn the procedures that ensure this interconnection.

Note that the Configuration server must have a connection with all other PortaSwitch® components via a private network interface. This network interface is also used by the PortaSwitch® servers to interconnect with each other.

To learn which ports must be open on PortaSwitch® servers, please refer to the *What is the recommended setup for PortaSwitch® behind a firewall?* section in the **PortaBilling® Installation Guide**.

For a detailed description of PortaSIP® component and processes, please refer to the *PortaSIP® Cluster Components* section in the **PortaSIP Administrator Guide**.

### Protocols and ports used by the RADIUS server

Subcomponent	Interacts with	Protocol	Ports	Details
<b>Billing Engine</b>	Main DB Server	TCP	3306/ 3307	MySQL Database server connection
<b>RADIUS</b>	PortaSIP® Cluster	UDP	1812/ 1813	Incoming RADIUS requests
<b>Alerter</b>	Main DB Server	TCP	3306/ 3307	MySQL Database server connection
<b>Disconnecter</b>	Main DB Server	TCP	3306/ 3307	MySQL Database server connection
	Other nodes (VoIP, WiMAX, WiFi)	TFTP / HTTP		
<b>Diameter</b>	Other nodes (VoIP, WiMAX, WiFi)	UDP		Incoming Diameter requests

## Protocols and ports used by the PortaSIP® cluster

PortaSIP® cluster components listen on following ports:

Subcomponent	Protocol	Ports	Details
<b>Dispatching node</b>			
EdgeProxy	UDP	5062	SIP port for control SIP traffic
EdgeProxy, SIP cluster protector	UDP	5060	production SIP traffic
EdgeProxy, SIP cluster protector	TCP	5060	production SIP traffic
EdgeProxy, SIP cluster protector	TLS	5061	encrypted production SIP traffic
Mail proxy	IMAP	8081	IMAP transport
Mail proxy	IMAPS	8091	IMAP over SSL
Mail proxy	SMTP	8101	SMTP transport
Mail proxy	UDP	5067	Mail Proxy control traffic
Processing Node controller	UDP	5063	Media Unit Controller UDP transport
Processing Node controller	TCP	5068	Telnet interface
Limit controller	UDP	5070	port for communication with the ProcessingNode controller
SMPP proxy	SMPP	2775	incoming SMPP connections
SMPP proxy	UDP	5064	port for communication with the ProcessingNode controller
<b>Processing Node</b>			
MWI	UDP	5264	The network port to bind the MWI daemon to
IMGate	SMPP	2775	incoming SMPP connection
IMGate	UDP	5960	port for incoming SIP SIMPLE messages
IMGate	TCP	5960	port for incoming SIP SIMPLE messages
IMGate	UDP	5961	port for outgoing SIP SIMPLE messages
IMGate	TCP	5961	port for outgoing SIP SIMPLE messages
IMGate	UDP	2775	port where the IMGate server creates a listening socket for SMPP connections
RTP proxy	UDP	35000–65000	RTP proxy port

<b>B2BUA</b>	UDP	5070	Base SIP port for B2BUA workers
<b>B2BUA</b>	TCP	5064	B2BUA telnet interface
<b>B2BUA</b>	TCP	5000	Call controller API
<b>B2BUA</b>	UDP	5059	B2BUA sibling communication
<b>Registrar</b>	TCP	5065	Registrar transport ports
<b>Registrar</b>	UDP	5065	Registrar transport ports
<b>Subscription manager</b>	TCP	5066	Subscription manager transport ports
<b>Subscription manager</b>	UDP	5066	Subscription manager transport ports
<b>IMAP server</b>	IMAP	143	IMAP transport
<b>IMAP server</b>	IMAPS	993	IMAP over SSL
<b>Log Master</b>	TCP	10000–10800	

### Protocols and ports used by the Web server

Subcomponent	Interacts with	Protocol	Ports	Details
<b>TaskStack</b>	Main DB Server	TCP	3306/ 3307	MySQL Database server connection
	Replica DB Server	TCP	3306/ 3307	MySQL Database server connection (SELECT requests only)
<b>Apache / FCGI</b>	PortaSIP® Media server / other nodes (VoIP, WiMAX, WiFi)	TFTP / HTTP		
	Main DB Server	TCP	3306/ 3307	MySQL Database server connection
	Replica DB Server	TCP	3306/ 3307	MySQL Database server connection (SELECT requests only)
<b>Cassandra</b>	Other nodes (VoIP, WiMAX, WiFi)	TFTP / HTTP		

## PortaOne Monitoring System

PortaOne software comprises a complex system that involves the simultaneous work of several servers or clusters of servers. Such a system requires reliable monitoring that makes it possible to immediately detect any nascent failure and prevent the possibility of negative consequences to the system. The *PortaOne monitoring system* collects and analyzes data from more than 100 types of indicators from each system and provides monitoring serviceability 24/7.

The PortaOne monitoring system provides the following functions:

- It continually collects information from each server within the entire PortaSwitch® installation.
- It provides fast detection of failures and informs the support team when a failure exists.
- It visualizes system performance via graphs.

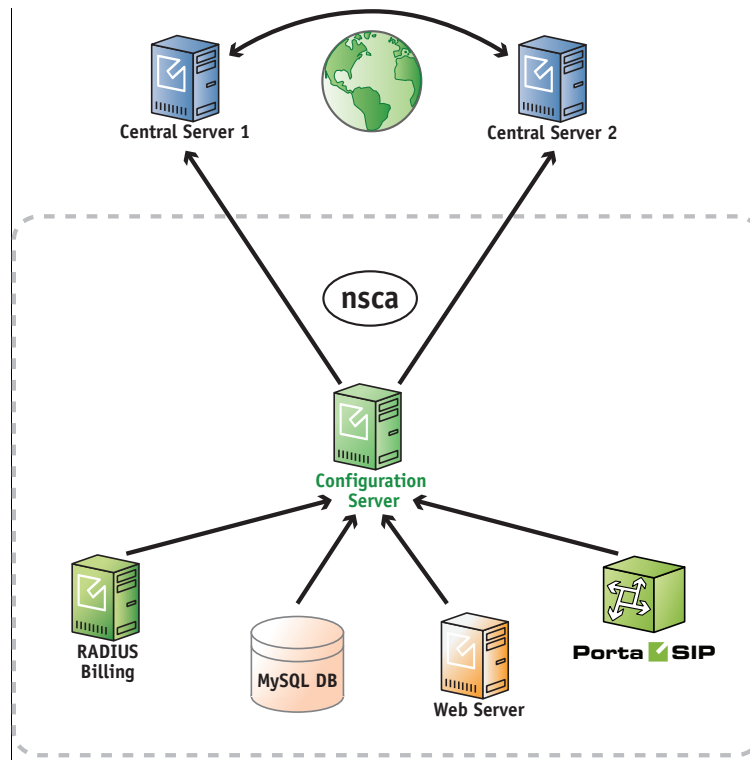
The *PortaOne monitoring system* is implemented through *Nagios* ([www.nagios.org](http://www.nagios.org)) – the industry standard in IT infrastructure monitoring. It offers complete monitoring and alerts for servers, switches, applications and services, and provides many additional plug-ins. The transport of data is realized through *Nagios NSCA (Nagios Service Check Acceptor)* packages. Nagios ensures a high level of process security and reliability and has been recognized for its efficacy by users all over the world.

### PortaOne NSCA-based Monitoring

NSCA-based monitoring is used on most customers' installations. There are three key entities involved in PortaOne NSCA-based monitoring (*Figure 1-1*):

- **Central Server 1 and Central Server 2** – These servers collect information from the ITSP's Configuration server about the status of each indicator for further analyses and processing. For security, these PortaOne® Inc. servers are physically located on different sides of the globe, while performing the identical work.
- **Configuration Server** – This server continually collects monitoring information from all of the PortaSwitch® Servers in the installation and sends it to the Central Servers.
- **PortaSwitch® Server** – This server's functions are displayed via a variety of indicators. PortaSwitch® Servers continually send information from all of the different indicators to the Configuration server.





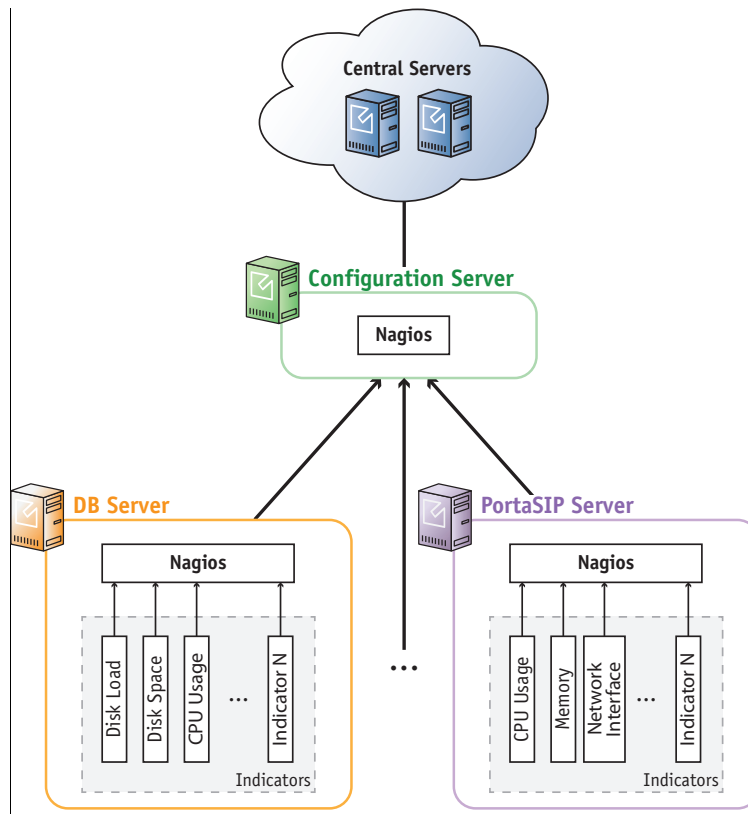
**Figure 1-1** – PortaOne NSCA-based monitoring

Very often data from one indicator is not sufficient for deciding whether the system is functioning correctly or not. Consequently, it is necessary to gather and analyze information from many different indicators to make a decision regarding relevant output. For this reason, decisions are made and enacted upon the Central Servers, since they collect and store all of the data from all of the PortaSwitch® Servers. The Central Servers assign the status for each indicator:

- **OK** – advises that the system is functioning correctly;
- **Warning** – warns about any type of abnormality in order to prevent system failure;
- **Critical** – informs that urgent intervention is needed for service outage or other failure prevention.

In order to transfer data from the indicators to the Central Servers, all of the PortaSwitch® Servers send data to the Configuration server via the private LAN segment. The interaction mechanism is the following (*Figure 1-2*):

1. Checks take place once per minute.
2. Nagios gathers check results.
3. NSCA-Client sends the results to the NSCA-Server.
4. Nagios goes back to waiting mode.



**Figure 1-2 – Nagios monitoring scheme**

Data from each indicator is sent separately. The Configuration server itself does not send requests to the PortaSwitch® Servers; it only passively receives the monitoring results. The main function of the Configuration server in this process is to reduce the load on the Central Servers. Therefore, it does not send each momentary instance of data continually, but rather, accumulates the data to send in bulk. When data from 20 indicators has been gathered, it is sent to the Central Servers within one single transaction. Similar to that of the Configuration server, the Central Servers do not send requests but instead, passively receive the monitoring results. When the results are received, the Central Servers assign a status – OK, Warning or Critical – to each indicator, and generates reports and statistics graphs.

Aside from collecting data from the PortaSwitch® servers, the PortaOne supervision system monitors data transportation. Nagios scripts collect data from each indicator once per minute. If, for example, a last message from a PortaSwitch® server is received more than three minutes previous or the data obtained by a Nagios script appears critical, Nagios will automatically generate a “Critical” status.

If, for example, PortaSwitch® is installed across multiple sites but due to a power outage, the servers at the main site became unavailable, the “stand-alone” mode is automatically activated on the secondary site so the

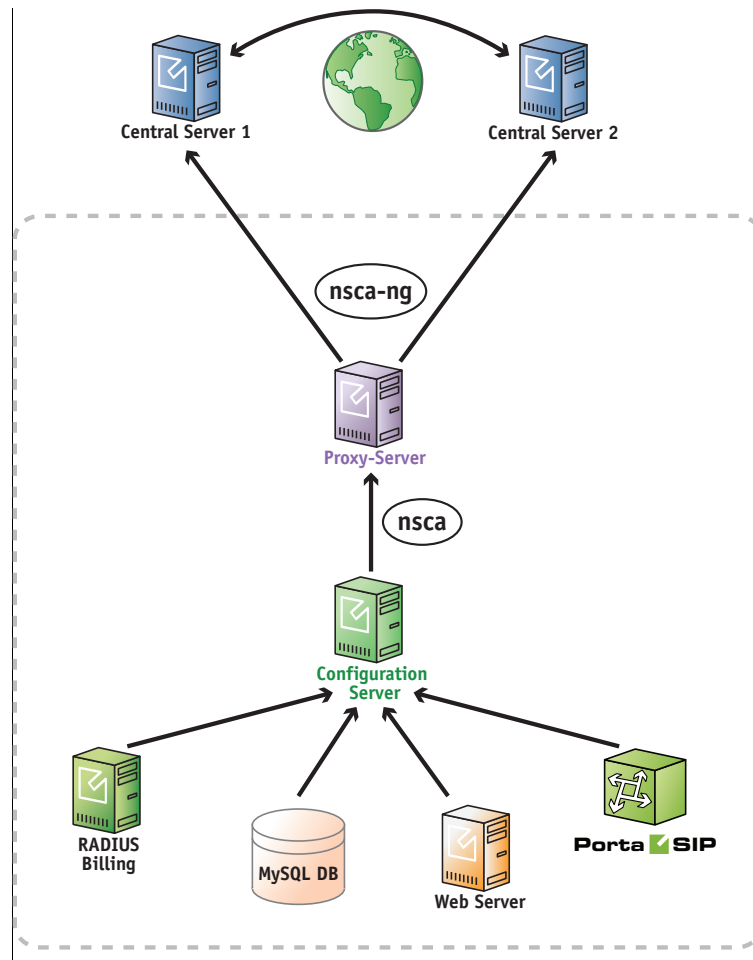
secondary site provides services to end users. Nagios on the Central Servers detects that data from the primary site servers is not reaching it and informs the PortaOne support team that urgent intervention is needed. The PortaOne support team immediately contacts the customer's team to address the issue. Once the primary site becomes available again, the PortaOne support team makes sure that all servers are functioning normally and that all corporate services are being correctly provided.

The PortaOne support team works 24/7 to assure that customers' installations are continually being monitored and supported even when the customers' own engineers are not in the office.

## PortaOne NSCA-NG-based Monitoring

Companies have different security policies. For example, some companies forbid connections from being established between their internal network and outside Internet hosts. This makes it impossible to send data being monitored from PortaSwitch® servers to the PortaOne monitoring system.

One solution we've come up with is to include an intermediate server within the company's DMZ zone. This was developed to resolve this issue, in particular (*Figure 1-3*). This server will proxy the data that's being monitored. The solution is based on the NSCA-NG package, which uses TLS encryption and shared-secret authentication, satisfying additional security rules.



**Figure 1-3** – PortaOne NSCA–NG-based monitoring

The process unfolds similarly to PortaOne NSCA-based monitoring. The differences begin when the Configuration server sends data to a customer's Proxy-Server instead of to the PortaOne Central Servers. The NSCA-server installed on the Proxy-Server receives the data and transforms the NSCA packages into NSCA–NG packages. The NSCA–NG client sends data that's encrypted with a secure password through the TLS protocol to the PortaOne Central Servers.

### Configuration

The PortaOne monitoring system is installed on each server within the PortaSwitch® installation with default settings. Then the PortaOne support team adjusts the settings as part of the post-install procedure. The monitoring system installation does not require any effort from the customer – unless they wish to take part in settings adjustment. For this, the list of all indicators' names and definitions can be downloaded from [here](#).

Some additional monitoring services can be provided to the customer upon request. For instance, the PortaOne support team can authorize a customer to receive notifications and alerts regarding Warning and Critical status by email.

PortaOne support also provides assistance if a PortaSwitch® customer wishes to implement their own additional script for monitoring. Please contact the PortaOne support team for more information.

## Updating the System to a New Version

Updating your system to a new release (whether it is your personal WiFi router or a powerful PortaSwitch® server, serving tens of thousands of customers) is always a challenging task. You ask yourself questions such as: How long will it take? Will updates be applied correctly to all system components involved? What happens if something goes wrong – can I get my system back to the operational state it is in now, etc.?

PortaSwitch® utilizes an innovative system of maintaining and modifying the software code on each server, allowing you to properly address all of the concerns expressed above and ensure that you are able to:

- Migrate quickly to a new maintenance release, without any problems on the way and obtain the system that operates 100% according to how it is intended to operate.
- In case there is something wrong with the functionality of the new release (e.g. you just realized that in order to properly use the new feature you need to train all of your staff, and this would take several days) you can safely rollback to exactly the same version of the software you were using prior to the update.

Most of the software systems currently used throughout the world contain a single copy of the “current” software so that when an update is done – some parts of it would be replaced with updated versions. This brings up a significant risk of incompatibility between the updated components – and the ones that remain from before the update may render the whole system unstable.

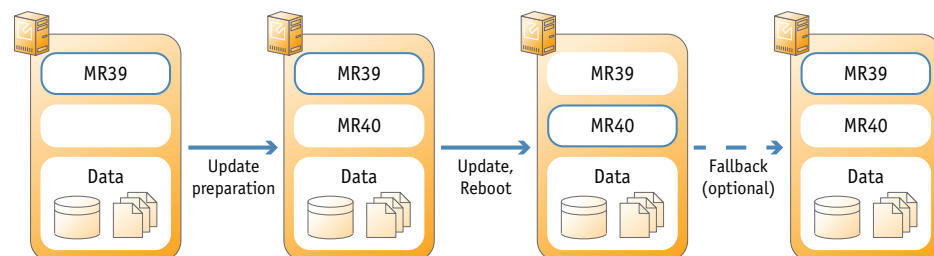
An alternative approach – when the entire code image is replaced with a new version (this is how you update the firmware in your router, for instance), poses the risk that if something goes wrong during the update process – the system ends up without any operable software and becomes totally unusable (my guess is that you probably heard about the iPhones which “brick” after an update?).

This is why PortaSwitch® utilizes a dual software version management system that has none of the weaknesses described above.

The disk subsystem on each PortaSwitch® server contains three (3) separate partitions. One of them is used to store the actual application data, i.e. database files, logs and .csv files with statistics of the customer's activity, etc. Two (2) other partitions are equal in size and each of them can contain the full set of the software “code” required to operate the server – operating system, third-party libraries and modules (e.g. Apache) and the actual code for a specific application, e.g. PortaSIP®. At any given moment one of these partitions is considered active: this means that upon startup, the server uses this particular partition to boot up and the application code, located within it, is used to operate the service. When the system is being prepared for an update to a new release, the other partition is cleared and the new version of the code is installed there. This is done while the system is still operating under the current version of the software, without any service interruption. Now the server has all the required data to operate with the new release – moreover, since the new release is installed as a set of binary packages, one can be sure that this is exactly the same code (the same version of operating system, the same version of kernel and the same bytes in every single utility or file!) that was used in PortaOne's labs during the testing period, that was deployed on staging systems during the field testing and that is currently being used by other PortaOne customers worldwide.

After that, the configuration agent updates the “local” files (e.g. “etc/hosts”) based on the system's configuration stored in the Configuration server: e.g. what IP address each service is working on and which application-specific features are on and / or off, etc.

Finally, at the specific time the new partition is marked as “active” the server is restarted using the new version of the code (these tasks are done automatically by the update agent, controlled by the Configuration server). The potential downtime is just a few minutes – the time required to complete the restart. Nothing is changed in the “old” partition though – so if a rollback is required, it only requires a reboot from that partition and the server is back to the old, “stable” release.



After some time when you wish to update to an even newer release, this partition is wiped clean and the new version of the code is loaded into the recently emptied location. Then the process described above repeats.

The same process is used to update to a new maintenance release or to a newer software build within the current release.

## Updating the Application Data

If all PortaSwitch® applications were “stateless”, in other words, if they were only doing some calculations based on the current input from the user and some pre-programmed rules – this chapter would end with the previous paragraph.

Actually, most of the applications in the real world, and PortaSwitch® is one of them, rely on a large set of previously accumulated data to make decisions about how the service should be provided. For instance, if the customer has already made calls to the US & Canada for a total of 101 minutes during this billing period, and his plan only allows for 100 free minutes – a call made right now would be charged at \$0.05/min rate.

All the data accumulated by the old software release are available to the new one after the upgrade to ensure the system’s proper operation – and this involves changing the data files, database structures and data to accommodate the new release.

So the update process includes two extra steps:

- Non-blocking data modifications are done as part of the “preparation” process, when the new release is already installed to the new partition, but before it becomes active. These modifications include adding new tables, inserting new records, etc. – basically anything that could be done while the system continues to operate with the old release.
- Blocking data modifications are those that may affect the system’s performance while they are being carried out. For example, adding a column to a table in MySQL would stop all other queries to that table from being executed until the operation is complete (another advantage offered by PortaBilling® Oracularius is that there are almost no “blocking” operations there). Blocking updates are done during off-peak periods. Needless to say, the PortaOne team tries to reduce the amount of blocking data structure modifications and the amount of time required for applying them.

The process described above allows the data modifications to be performed while the system still operates using the current release. There is one important consideration, though: during that time, the “older” version of the release operates with the “newer” version of the data. (The same situation would happen if you were to rollback to the older release).

The PortaOne team has a development and testing process specifically aimed to make this possible, but we can only guarantee this inter-

operability for the adjoining releases. In other words, the system operating with MR39 will operate normally while the data is being updated to MR40 (or it is possible to rollback to MR39 after the migration to MR40 has been done). It is not possible to provide this transparency when the distance between releases is too great (e.g. MR36 will most likely *not* operate properly if the data has already been updated to the MR39 format). Thus, the preferred method of updating that provides unlimited time for update preparation and allows for fallback to the previously used version is to go step by step: e.g. from MR39 to MR40, then from MR40 to MR41 and finally from MR41 to MR42, etc. (If required, it is still possible to do a MR36 to MR39 update in one go, of course – but in this case there is no possibility of performing a rollback).

## Zero-Downtime Update

As a result of PortaOne's agile development cycle, new maintenance releases are delivered every 2 months, and each new release contains numerous enhancements that enable new or improved services. Naturally telco operators want to maintain an up-to-date system to uphold their competitive edge – and this has to be done without any negative impact on the end user.

Zero-downtime update (ZDU) technology allows you to perform software upgrades at any time without any noticeable impact on end users.

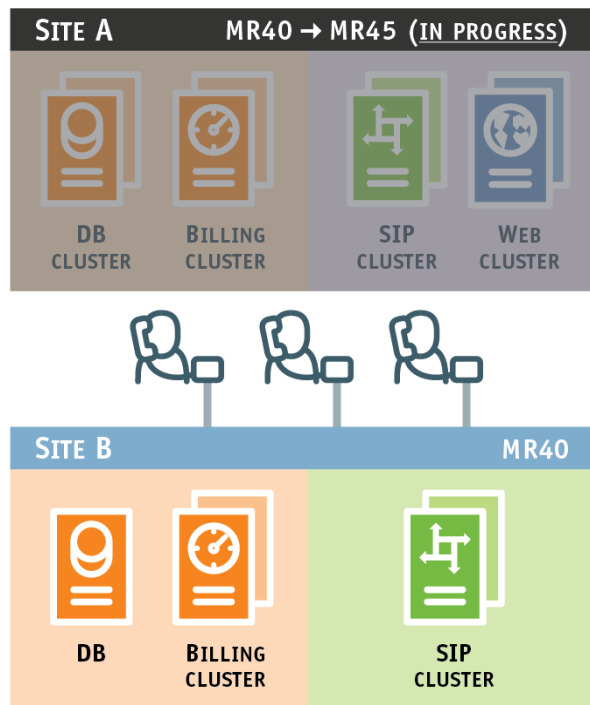
ZDU utilizes PortaSwitch site redundancy architecture – at least two sites are required. The upgrade is performed via this following procedure:

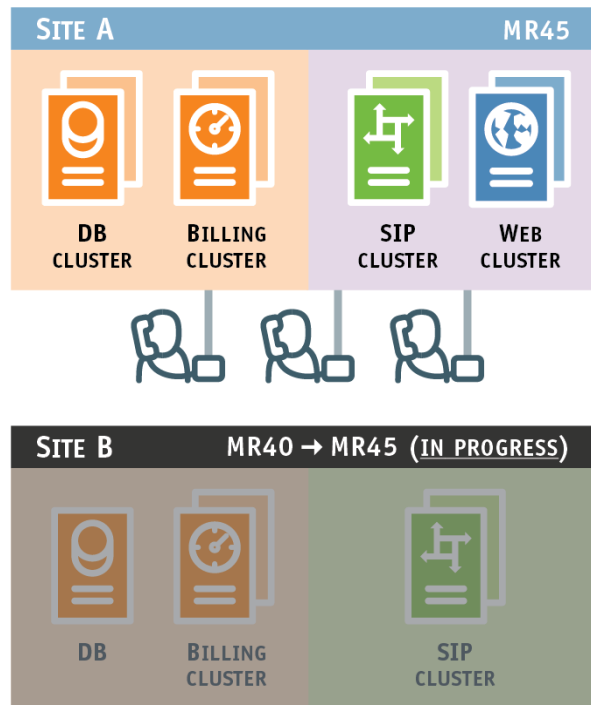
- First, the usual upgrade preparation is done (verification of custom modifications, hardware check for compatibility with new OS, etc.).
- Then secondary site B is switched to the stand-alone mode. All new client traffic is relocated to site B (done via modification of DNS records, change of routing on the external SBC and altering IP routing, etc.).
- After all existing calls or sessions are completed on site A and there is no live traffic there, the upgrade process begins on that site. Database changes are applied, a new version of code is installed and servers are rebooted.
- Throughout this time, customers are able to use services via site B as usual; calls / messages / sessions are charged, balances are updated accordingly and xDRs are being written. The only restriction during this time is that administrators or customers cannot change their service configuration via the web, etc.
- Finally, when the upgrade is completed on site A and all services there are verified to be working properly, site A is activated. Then the synchronization process with site B begins, and changes in



balances and xDRs accumulated on site B while in stand-alone mode are applied to the main database on site A.

- All new customer calls / sessions are directed to site A (using the same tools as described earlier). Any calls in progress on site B will complete as normal.
- When site B has fully synchronized its data with site A and there are no more “live” calls there, the update process begins on site B. It replicates all changes in the database structure from site A, a new version of the software is installed and servers are rebooted.
- Finally, site B returns to its standard operational mode: PortaSIP servers there may be used in conjunction with the main site to process calls; the billing servers and database are in stand-by mode and data changes on site A are immediately replicated to site B so it has an up-to-date copy of all service configurations.





The scenario described can be used with multiple sites (e.g. in different countries), not just with two. The procedure is exactly the same – the only difference is that secondary sites will be updated one by one or in groups.

As a result, even if the upgrade process takes several hours, customers are able to access the complete service at all times and there is no “visible” downtime.

## Custom Modification Management

To compete with larger incumbent telco operators, independent service providers must not only provide the functionalities that end users want faster than anybody else on the market – but also provide these functionalities exactly how the end users anticipate seeing it. The first condition is perfectly implemented by PortaOne and its agile development process, which allows PortaOne to offer an incomparably short waiting time for a new functionality. The second condition can be achieved by adjusting standard functionality according to the unique needs of your customers.

Modifications to standard functionality can be both light and solid. But in both cases you will need to introduce changes to the software installed on your servers and maintain them through software upgrades. Usually, the more modifications you introduce, the more difficult they become to

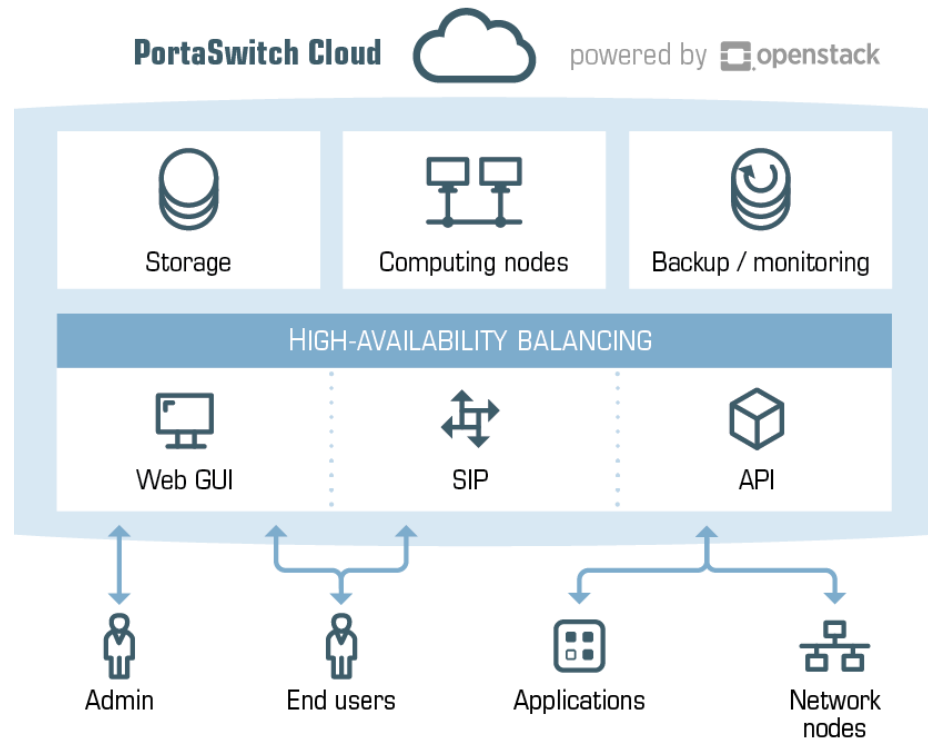
maintain. To simplify this process, PortaSwitch® provides convenient tools for managing your custom software, patches and files:

- **Custom software** – You can upload new third-party RPM packages to any server within the installation and keep track of their status and versions.
- **Patches** – You can add patches to both PortaOne and third-party RPM packages to make sure that patches are automatically applied to a new release after a software upgrade. Moreover, it is possible to define a patch’s “lifetime” to automatically stop its propagation with an upgrade to a specific software release or build.
- **Files** – You can create a list of custom files (e.g. sudo configuration files) and directories that must remain on your servers during the software upgrade.

These tools allow your development team to automate the management of custom modifications and shift a significant amount of this work to PortaSwitch®. Please refer to the [PortaBilling® Configuration Server Web Reference Guide](#) for more information.

## PortaSwitch® Deployment in a Cloud

PortaOne introduces the concept of PortaSwitch® as a cloud-based solution. With this solution, you do not need to purchase hardware servers and /or spend time and effort on their maintenance. Instead, you decide upon the processing capacity you require for your business, submit your request to the PortaOne Sales team and then receive a scalable, ready-to-use platform that operates in a cloud. Thus you start your business faster and speed up your payback.



The key features of the cloud-based PortaSwitch® solution are:

- Near-zero CAPEX and reduced OPEX – By deploying PortaSwitch® in a cloud, you save on hardware purchases and provisioning. The cloud infrastructure is maintained by PortaOne, which drastically reduces the load on your administrative staff.
- Reduced deployment time – You receive the platform, ready-to-use, within a day.
- Flexible scalability – The processing capacity can be increased / decreased on demand.
- Cloud pricing – The cloud-based PortaSwitch® solution does not assume licensing. The pricing consists of a one-time set-up fee and then a recurring subscription fee that is formulated based on the maximum number of concurrent calls and billable xDRs processed per month. Customers interested in migrating their existing deployments to a cloud will be eligible for commercial incentives.

The cloud-based PortaSwitch® platform is the solution for businesses of all sizes. Due to a reduction in deployment charges, small telcos and startups can use it to enter the telecommunications market. Large ITSPs can also use PortaSwitch® to:

- Deploy a new installation (site) in another region / country to introduce geo-redundancy and perform software upgrades with zero down-time;

- Gradually move active customers to new installations using the Porter data transfer tool and thereby introduce a dual-version of PortaSwitch®. (For more information about Porter, please read the [Moving Billing Data Between Installations](#) section);
- Quickly deploy a staging system to verify new functionalities or interoperability with third-party equipment.

With PortaSwitch® deployed in a cloud, you have advanced flexibility in managing your network infrastructure. Even with reduced deployment budgets, businesses of all sizes can become active players in the telecommunications market.

# **2. Integration with Third-Party Systems**

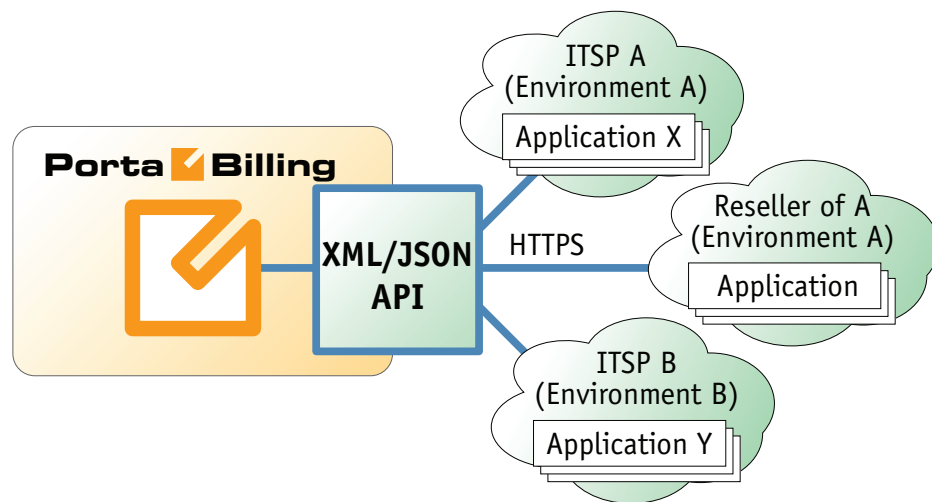
## Overview

PortaSwitch® is a system with an open architecture. Our main aim is to enable service providers to easily integrate PortaSwitch® into their network, to facilitate interconnection with third-party applications, and to simplify day-to-day tasks such as new customer activation or rate management.

## API for Data Operations

Although it is possible for an external application to access billing data directly in the database, PortaBilling® allows you to perform operations such as data retrieval or data modification via API using the following Web Application Services: XML (SOAP) and JSON RPC.

This is ideal for applications such as external web portals (where you only need to create a front-end to present the data to the end user) or order entry / provisioning systems (where an application needs to supply the new customer's data to PortaBilling in order to activate him).



Performing operations via API has several advantages:

- It is based on either XML/SOAP (Simple Object Access Protocol) or JSON (JavaScript Object Notation) Web Application Services and HTTPS transport, so it is accessible from any platform or operating system, and all communication between the server and clients is secure.
- The business logic embedded into the API provides integrity checks for all data modifications, and can compile records from several database tables to create a single customer information retrieval structure.
- PortaBilling® API is accessible to every owner of a virtual environment or reseller. Each user's access is automatically limited

to his “visible” portion of the available data, e.g. a reseller can only retrieve information about his own sub-customers or their accounts.

PortaBilling® XML / JSON API allows users to perform select, update, insert or delete operations on entities such as customers or accounts. Each user has his own login credentials, and each operation he wishes to perform is analyzed to determine if it is possible with regard to general data integrity (e.g. a new account cannot be created without being assigned to a customer) as well as the particular user’s security permissions (ACLs) (e.g. while it is possible in general to create new accounts, this user may be prohibited from doing so).

Details on XML / JSON API (such as available methods and data structures) are described in the [PortaBilling® XML / JSON API Reference](#).

## API for Computer Telephony Integration Services

CTI (Computer-telephony integration) technology enables users to control and monitor calls from external applications (e.g. CRM, web-based console, etc.) as if managing them from an attendant console. For example, a user can start a call from a CRM computer application and check the list of active calls for a certain group of users, etc.

The PortaSwitch® API includes a range of methods for integrating external applications with PortaSwitch® and performing remote call control.

These methods enable an API user to originate, answer, terminate a call, retrieve a list of currently established calls and subscribe to notifications about call state changes for certain customers and accounts. Together with already existing API methods (e.g. for retrieving customer information), these help build a full-grown CTI solution.

**NOTE:** File upload / download is not yet supported.

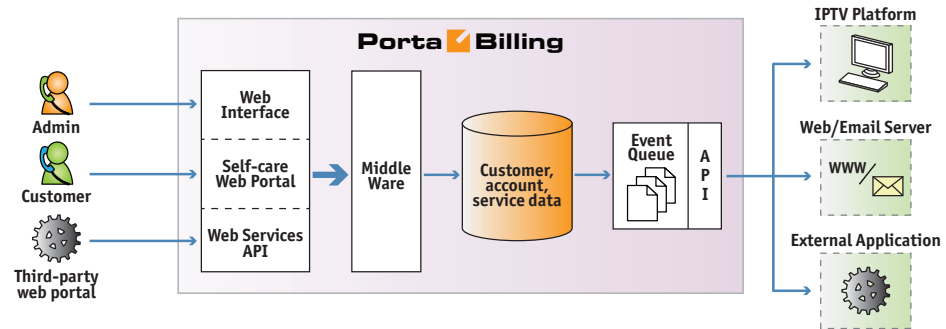
## Provisioning of External Systems

To simplify integration with external systems (e.g. IPTV platform or website hosting server) which receive their service configuration from PortaSwitch®, a dedicated interface is created so that all provisioning tasks can be controlled and managed from a single location.

Every modification of an object such as an account or customer in PortaBilling is recorded as an event. These events are queued in the



system and then an updated service configuration for each account is pushed out to one or several provisioning plug-ins. Each of these plug-ins provides an interface for supplying data to a specific external system. This could be a text configuration file for a legacy application, or an XML API provisioning interface for a state-of-the-art service platform.



The extensible framework allows service provisioning for new platforms to be done quickly and with minimal effort.

# 3. Dual-Version PortaSwitch®

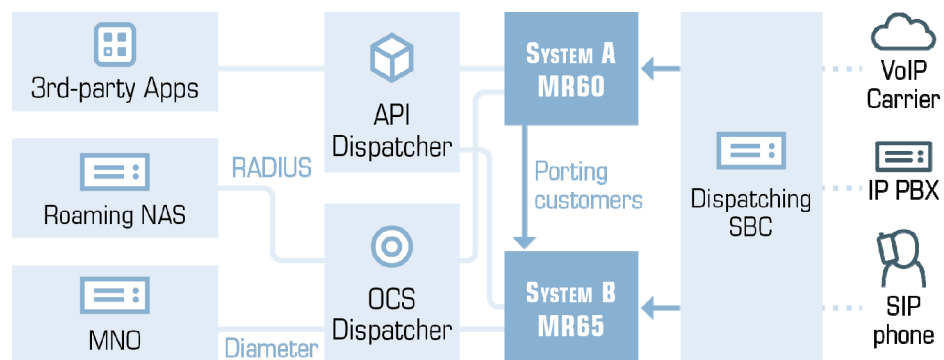
## Concept of Dual-Version PortaSwitch®

Dual-version PortaSwitch® enables service providers to balance between having access to new features and securing the overall stability of their existing platform. It comprises two systems of current release (e.g. MR60) and new release (e.g. MR63) which operate simultaneously.

Prior to moving services to a new system (e.g. from staging to production) or a new release (e.g. from MR55 to MR60), you may want to test the service flow within the new billing environment for your existing customers. To eliminate human error during service configuration and ensure its correctness, PortaSwitch® has a data transfer tool.

### Moving Customer Data in Dual-Version PortaSwitch®

Using Porter functionality, you can transfer some of your existing customers to the new system and immediately run services there with minimum reconfiguration efforts.



Porter is a script with which you export a customer's service and billing configurations and all of the dependent entities (accounts, products, service features, xDRs, invoices, etc.) from a previous "source" system and import them into the target.

**NOTE:** You must have the Root access level to perform the data transfer.

As a result, the following data is transferred:

Entity	Transferred Data
Customer	<p>Customers' configuration:</p> <ul style="list-style-type: none"> <li>Customers and their accounts.</li> <li>Customer and account service features.</li> <li>xDRs.</li> </ul>

	<ul style="list-style-type: none"> <li>• Invoices.</li> <li>• Payment methods.</li> <li>• Customer sites.</li> <li>• IP Centrex elements – extensions, huntgroups and call queues.</li> <li>• Customer credit limit and payment history.</li> <li>• CPE and SIM inventory records.</li> </ul> <p style="text-align: center;">Generic configuration:</p> <ul style="list-style-type: none"> <li>• Custom access levels.</li> <li>• Customer classes.</li> <li>• Customer products and product groups.</li> <li>• Customer tariffs and rates.</li> <li>• Spending plans.</li> <li>• Subscriptions.</li> <li>• Volume discount plans and counters.</li> <li>• Dialing rules.</li> <li>• Fraud traffic profiles.</li> </ul>
<b>Vendor</b>	<ul style="list-style-type: none"> <li>• Vendors and connections.</li> <li>• Vendor DID batches.</li> <li>• Vendor tariffs and rates.</li> <li>• Vendor xDRs</li> </ul>

By default, customer xDRs that pertain to the current billing period are transferred. However, you can reconfigure Porter to transfer all of a customer's xDRs, if necessary.



**TIP:** To enable customers to use the services in the new system sooner, you can perform two-phase data transfer: first transfer customers without their xDRs. Then, transfer these customers' xDRs with the second run of Porter.

**Reseller** data is similarly transferred: the reseller configuration and that of their subcustomers and subresellers is transferred.

To preserve bilateral traffic exchange for **vendors**, pre-configure them as customers on the target system.


For the administrator's convenience, the following **general configuration** data can be separately transferred (either assigned or not to a customer or an account):

- Products;
- Subscriptions;
- Volume discount plans;

- Tariffs and rates;
- Customer classes.
- DID pricing batches and DID groups
- Custom field names.

**NOTE:** Customer .csv files with xDRs and voicemail messages are not transferred by Porter.

Porter operates in interactive mode. This means that if during a data transfer some entity appears to already exist in the source system (e.g. you transferred the product earlier), you may choose to use this entity or create a new one.

When transferred, a customer and their accounts acquire the  **Exported** status in the source system. This means that customer service provisioning and billing is stopped in the source system. As soon as the administrator imports the customer, it resumes in the target system.

To reduce this downtime, during which a customer is “in-between” systems and cannot use the services, Porter creates a lineup of entities to transfer and processes them one by one. In other words, it exports customer A’s data from the source system and immediately imports it to the target. Then it repeats that same procedure to transfer customer B, etc.

**NOTE:** If customer accounts were registered on the PortaSIP® server during the data transfer, they are displayed as registered until the time of their registration expires.

For the data transfer to take place, the following entities must be pre-configured in the target system:

- A PortaBilling® root user;
- Currencies;
- Destinations;
- Nodes;
- Templates (invoice, tariff upload / download);
- Destination groups and destination group sets;
- Off-peak periods.

These functionalities must also be pre-configured in the target system if a customer or an account uses them:

- Bundle promotions.
- Routing plans.
- Routing categories.
- Routing criteria.
- Service policies.
- Internet access policies.

- Geo-risk profiles.
- CPE profiles
- Fraud traffic profiles.



Any custom configuration that is defined in the source system (e.g. custom notification templates) must be pre-configured in the target system as well.

Porter functionality helps you evaluate the service flow under the new conditions and control the migration process in the most preferable timeframe and pace.

## Call Delivery in Dual-Version PortaSwitch®

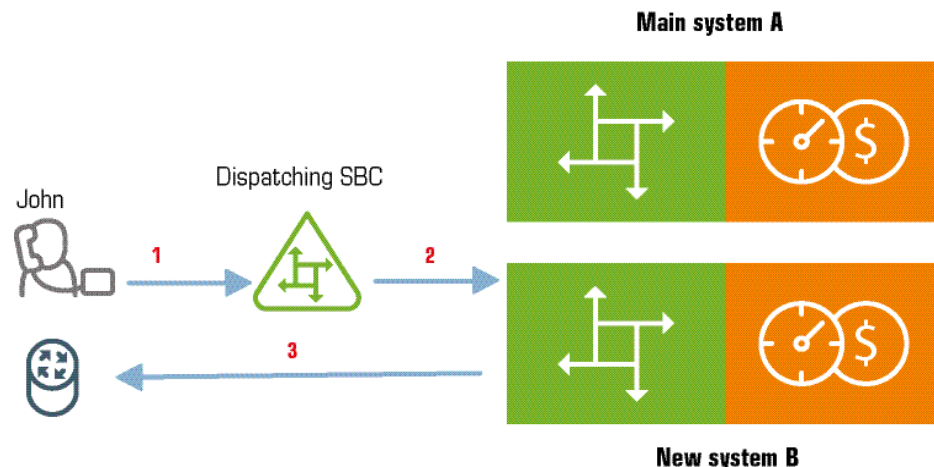
For the successful operation of dual-version PortaSwitch®, it is essential that customers do not notice being moved to a new system. That is, if John Doe is already in the new system, he must be able to make and receive calls as if nothing has changed.

To deliver calls to accounts in either system PortaSwitch® uses the dispatching SBC. This is a dedicated PortaSIP® node that operates as the inter-system proxy and dispatches call initiation requests across systems. It operates in high-availability mode and presents a single point of entry to both your termination partners and your wholesale, SIP trunking, residential and other customers.

### Usage scenarios

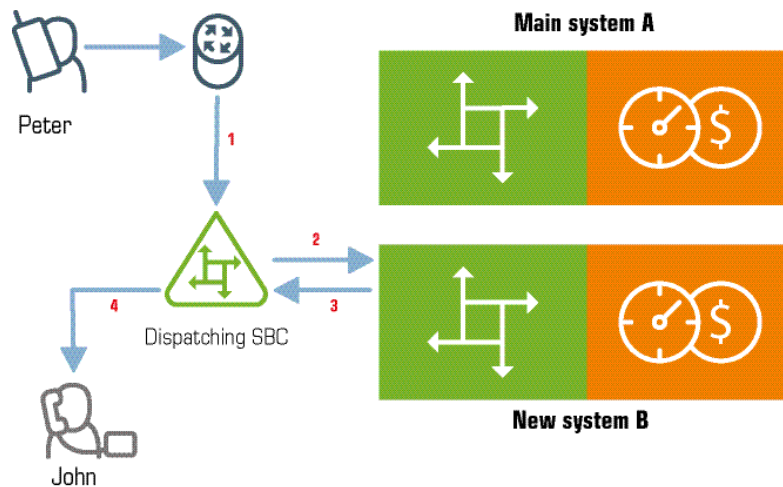
Let's have a closer look at how PortaSwitch® processes calls for John who the administrator transferred from main system A to new system B. For both making and receiving calls, John's phone is registered with the dispatching SBC's IP address.

When John calls his friend Harry in the UK, the following occurs:



1. The call request from John's phone arrives to the dispatching SBC (1).
2. The dispatching SBC looks up John's account in the database and detects that it is provisioned in system B.
3. The dispatching SBC routes the call to system B's PortaSIP® (2).
4. PortaSIP® sends the authorization request to the respective PortaBilling®.
5. PortaBilling® authorizes the call and provides PortaSIP® with the routing results.
6. PortaSIP® sends the call to the vendor to establish the call with Harry (3).
7. When the call finishes, PortaSIP® sends accounting information to PortaBilling® to calculate the charges and produce xDRs.

If someone calls John, the call processing procedure is quite similar. When Peter dials John's number:



1. The vendor receives the call from the PSTN network and sends it to the dispatching SBC (1).
2. The dispatching SBC looks for the DID number that matches the destination number in the database.
3. The dispatching SBC finds that the destination number corresponds to John's account and is provisioned in system B.
4. The dispatching SBC routes the call request to system B's PortaSIP® (2).
5. PortaSIP® sends the authorization request to the respective PortaBilling®.
6. PortaBilling® authorizes the call. Since the call is intended for the local account, it instructs PortaSIP® to send the call to the dispatching SBC (3).
7. The dispatching SBC delivers the call to John's phone and John and Peter start their conversation (4).

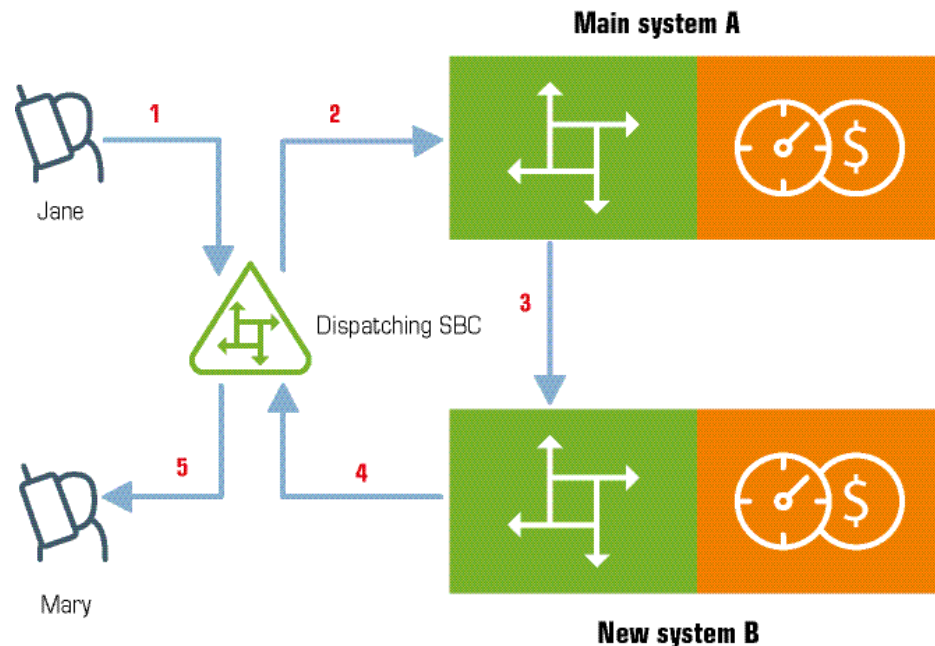
8. Once the call finishes, PortaSIP® in system B sends accounting information to PortaBilling® to calculate the charges and create an xDR for John.

### Inter-system calling

When customers are moved from one system to another, they must be reachable both from the external network and from the previous system. One way to deliver calls to such customers is to use the PSTN network. However, this is impractical since it adds hops in the call path and imposes additional charges.

To optimize call delivery and eliminate excessive charges, dual-version PortaSwitch® has the inter-system routing mechanism as the direct path for calls throughout the whole network.

Calls between systems are delivered using special rate codes. These codes represent the billing environments of either system and are derived from their unique IDs. This distinguishes this routing system from intra-system routing.



Thus, if Jane, from the main system, calls Mary, who is already in the new system, the following occurs:

- The call from Jane's phone arrives to the dispatching SBC (1)
- The dispatching SBC sends a call request to PortaSIP® at the main system for processing (2).
- The main system's PortaSIP® authorizes the call in PortaBilling®.



- PortaBilling® detects that the destination number belongs to the account that has been moved to the new system and therefore instructs PortaSIP® to send the call to the new system.
- PortaSIP® on the main system routes the call to PortaSIP® on the new system, bypassing the dispatching SBC (3).
- Upon call authorization in the new system's PortaBilling®, PortaSIP® sends the call to the dispatching SBC (4), which delivers the call to Mary's phone (5).
- When the call finishes, each PortaSIP® instance sends accounting requests to PortaBilling®. Then the main system's PortaBilling® creates an xDR for Jane's account while the new system's PortaBilling® creates an xDR for Mary's account.

## Proxying Diameter requests in Dual-Version PortaBilling®

In mobile networks, PortaBilling® operates as the OCS (Online Charging System) and communicates with the EPC via the Diameter protocol. In this deployment, the Diameter cluster serves as the point of entry to the OCS, accepting and processing Diameter requests from the SAE-GW.

To ensure uninterrupted service provisioning during gradual customer migration, the Diameter cluster on the new system serves as the inter-system proxy. It accepts all Diameter requests from the main system to the new one in dual-version PortaBilling® and detects where in the system the account is currently located. If the account belongs to the new system, it processes the Diameter request itself. If the account belongs to the main system, the Diameter cluster proxies the request to the main system's Diameter cluster and mediates further communication with that Diameter cluster and the SAE-GW.

This solution facilitates customer transfer from the main system to the new one and ensures proper functioning of the dual-version PortaBilling®.

## Common API Entry Point for Dual-Version PortaSwitch®

Dual-version PortaSwitch® is a solution that enables service providers to perform smooth and controlled customer migration. It also provides extensible API for integration with third-party applications, CRM systems, building self-care portals, etc. Users must be able to log in to the same portal, under the same address regardless of their current location within dual-version PortaSwitch® in order to preserve their customary habits.

Thus, John Doe will log in to [www.mybilling.com](http://www.mybilling.com) to check his balance and invoices, both when his record is on the main system and after it is moved to the new one. Administrators and resellers, in turn, must be able to operate in both systems, via the API, without reconfiguring their applications.

To this end, the API dispatcher for dual-version PortaSwitch® serves as a single API entry point that unites both systems. The API dispatcher accepts API requests from applications and dispatches them within dual-version PortaSwitch® for processing. Thus, an application can receive data from the main, the new or even both systems. The decision about which data to retrieve is based on the following:

- Who uses the application – the administrator / reseller or a retail customer / account, and
- The user's location – if it is within the main or the new system.

Let's consider how retail customers and administrators / resellers operate in dual-version PortaSwitch® via the API, separately.

### **API for retail customers / accounts**

Customers and accounts can only operate with a system where their records are located. Thus, when a customer logs on to the self-care portal, the application sends the API request to the main system. The API dispatcher checks the customer's location within dual version PortaSwitch® and if the customer was moved to the new system, it establishes the session with the new system.

Then, when the customer performs some action (e.g. selects xDRs for the previous billing period), the API dispatcher proxies the API request to the new system. The new system retrieves the xDRs and the API dispatcher delivers them to the application.

If a customer on the main system uses the self-care portal, their API requests are processed by the main system.

### **API for administrators and resellers**

Administrators, resellers and their staff (representatives, customer care staff, etc.) manage their own configurations plus also the customers who exist within both systems in dual-version PortaSwitch®. Therefore, depending on customer location, their applications must be able to send the proper context, i.e. which system will process their requests.

For unambiguous identification of customers and accounts during the migration process, the systems in dual-version PortaSwitch® are using the shared registry for customer / account records creation. As a result, the IDs are unique across both systems so either a customer or an account is

always identifiable by their ID. For detailed information and assistance in record registry configuration, contact PortaOne Support.

After an administrator has configured the interconnection between the systems, the typical workflow is the following:

- The application connects to the main system via the API and receives the session ID.
- If the application sends the API request to retrieve the customer list, the API dispatcher runs the request in both systems and merges the results within a single list. Thus, the administrator or reseller sees those customers who are still within the main system, those who were moved to the new one plus those created in the new system.
- To manage a customer from either system, the application sends the ID of this customer's record within an API request. The API dispatcher finds the system where this customer is provisioned and runs the request there. After results are received, the data is delivered to the application.
- Entities such as products, bundle promotions, volume discount plans, etc. can exist independently, i.e. not be tied to a particular customer directly. Therefore, to retrieve the list of subscriptions from the new system, the application sets the session context by providing the unique ID for the new system's billing environment within the API request. Then the API dispatcher runs further requests within the new system.
- To operate with subscriptions from the main system, the application switches the session context by providing the ID of the billing environment in the main system. In this case, all subsequent requests will be processed by the main system.

The API dispatcher for dual-version PortaSwitch® provides a single place for customer management and system operation and smoothes the migration process. The full application preserves user habits and thus improves the overall customer experience in PortaSwitch®.

### Implementation specifics

When operating with dual-version PortaSwitch® via the API, consider the following implementation specifics:

1. Applications can operate with PortaSwitch® only via the REST and SOAP API. The WebSocket API is not supported.
2. To establish a session with the new system, credentials for the API access must be the same for both systems.
3. After login, the application will be provided with the session ID that must be used in all subsequent API requests.
4. All communication between the application and PortaSwitch® is done via the main system.

5. The common API entrance point operates in conjunction with the dispatching SBC and Diameter proxy.
6. Only the `get_customer_list` method provides results from both systems. If you have defined limits for the list (the number of rows to retrieve), expect results that are twice as long because the same limit value will be used when querying both the main and the new system.

### ***Known limitations***

Bear in mind the following known limitations:

1. Since the application communicates only with the main system, information (e.g. its IP address) about the application on the new system is not available.
2. You can restrict access by IP address only on the main system. Since the main system connects to the new one, IP access restrictions on the new system are not considered.
3. The systems are mapped with each other by their environment IDs. Therefore, be careful when switching the environment since it may result in broken mapping and session disconnect with the new system.
4. If the main system fails to establish the API session with the new system, it operates as if there is no new system.