PORTA ONE

PortaSwitch

Architecture and Concepts

Copyright notice & disclaimers

Table of Contents

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/**.

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) Servers for PortaSwitch® or nine (9) servers for PortaSwitch® Clustered. 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 website: **https://linux.oracle.com/pls/apex/f?p=117:1**

Client system recommendations

- **OS:** MS Windows 7 or above, Linux/BSD, macOS Sierra (version 10.12) or above.
- **Web browser:**
    - Google Chrome 84 or above, Mozilla Firefox 78 ESR, Mozilla Firefox 80 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 1366 × 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 minutes/5 minutes/30 days = 66,666 calls and then round up this number to 70,000, for convenience.

- *PortaBilling®:*
    - Calls: 70,000 calls per day * 50 KB * 2 ≈ 7 GB for one day
    - Registrations: 288 registrations/day per phone * 10,000 phones * 5 KB ≈ 14.4 GB for one day
    - Total for one day: 7 GB + 14.4 GB = 21.4 GB
    - Total for three days: 21.4 GB * 3 days ≈ 64.2 GB

- *PortaSIP®:*
    - Calls: 70,000 calls per day * 100 KB * 2 ≈ 14 GB for one day
    - Registrations: 288 registrations/day per phone * 10,000 phones * 10 KB ≈ 28.8 GB for one day
    - Total for one day: 14 GB + 28.8 GB ≈ 42.8 GB
    - Total for three days: ≈ 42.8 * 3 days ≈ 128.4 GB

- *The centralized log storage:*
    3 days * (21.4 GB + 42.8 GB) + 7 days * (21.4 GB + 42.8 GB) * 5% ≈ 192.6 GB + 22.5 GB ≈ 215 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 calls per day * 1.5 KB * 3 CDRs * 60 days * 2 * 2 ≈ ≈ 75.6 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
		keep logs for last two days only		
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. OCS 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 **PortaSwitch Installation Guide**.

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, PortaSIP® 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 PortaSIP® 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.
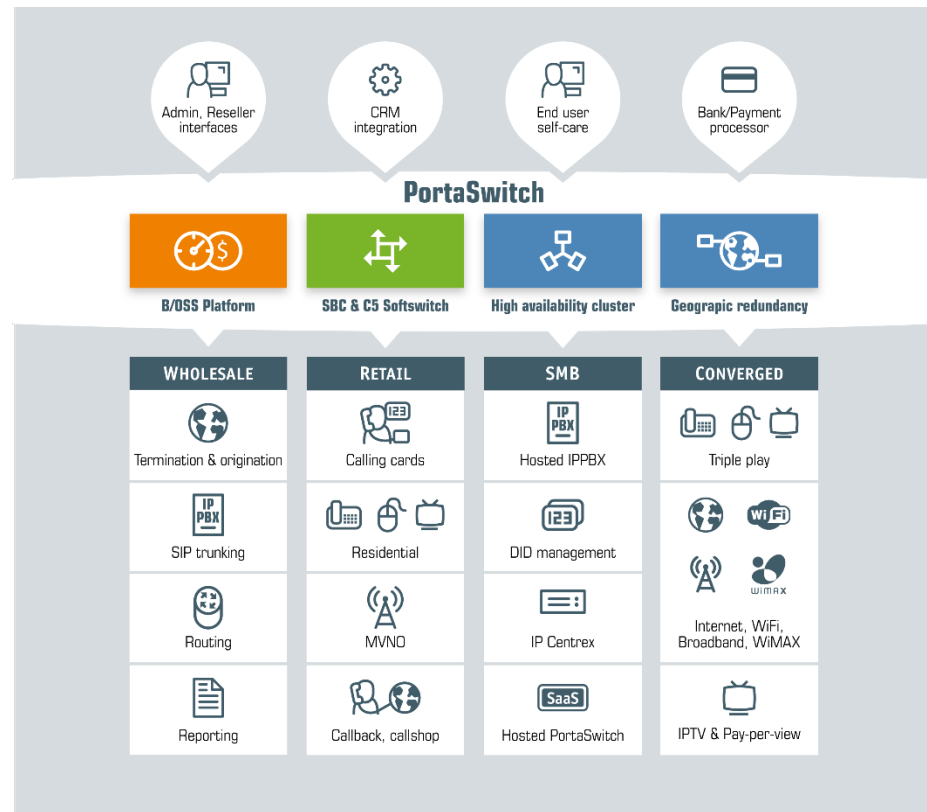
Centralized configuration management

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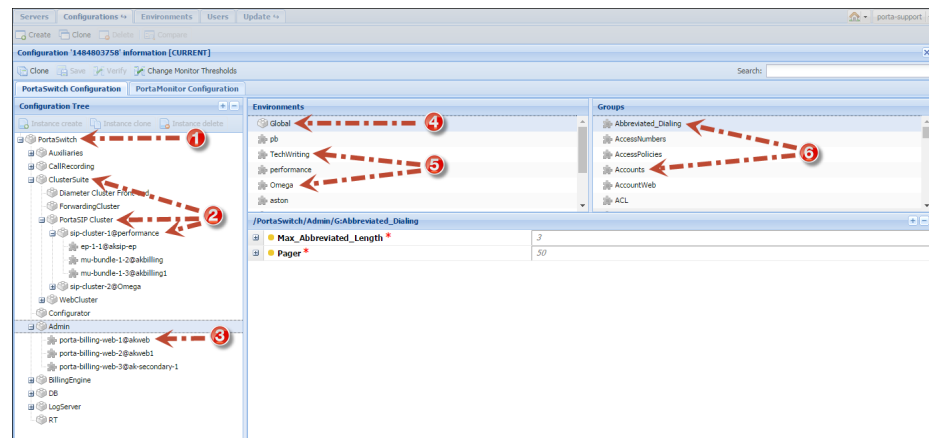
There are global options at the top level of the tree (**PortaSwitch** (1)), i.e. those that have an effect on all of the components of the system.

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**.
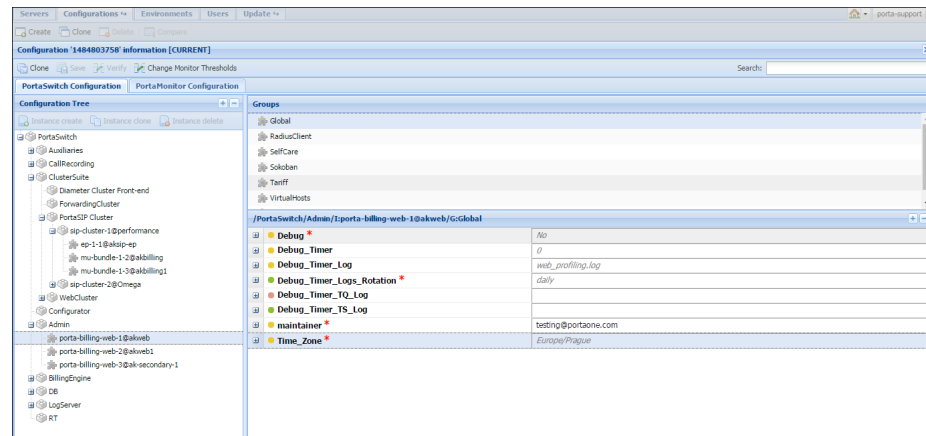
Some options may have different values in different virtual environments. These options are organized into environment sets, and each set provides control for options that are specific to a particular virtual environment. Virtual environments have their own hierarchy; the **Global** set (4) is the highest 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).
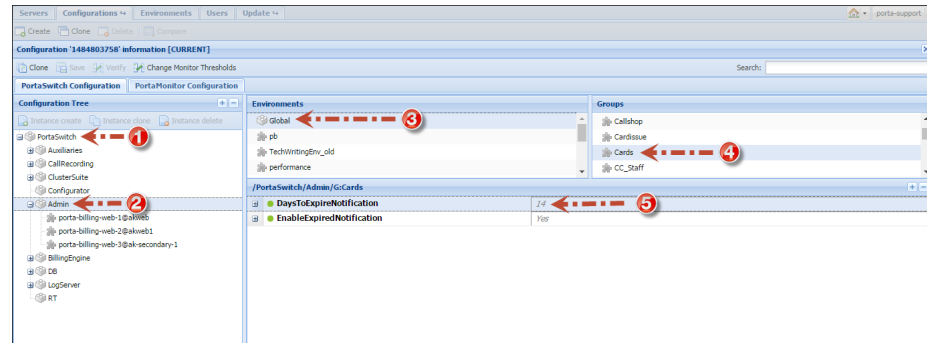
This value then becomes the default value for all of the virtual environments (those that exist and those not yet created), and PortaBilling® will send notifications to customers about their card expiration 14 days before the expiration date.

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® Clustered installation license covers nine 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® Clustered on only eight of their servers:

№	Server name	Running PortaSwitch applications
1	Alpha	• Billing engine
2	Beta	• Billing engine • Master DB
3	Gamma	• Web server
4	Epsilon	• Web server • Replica DB
5	Zita	• PortaSIP® dispatching node
6	Iota	• PortaSIP® dispatching node • PortaSIP® processing node

7	Sigma	• PortaSIP® processing node
8	Omega	• Configuration server

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 OCS 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 OCS 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 OCS service 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
<?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>
```

How does it work?

You can make changes to your "live" system at any time (e.g. add a new PortaBilling® OCS 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 OCS 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.
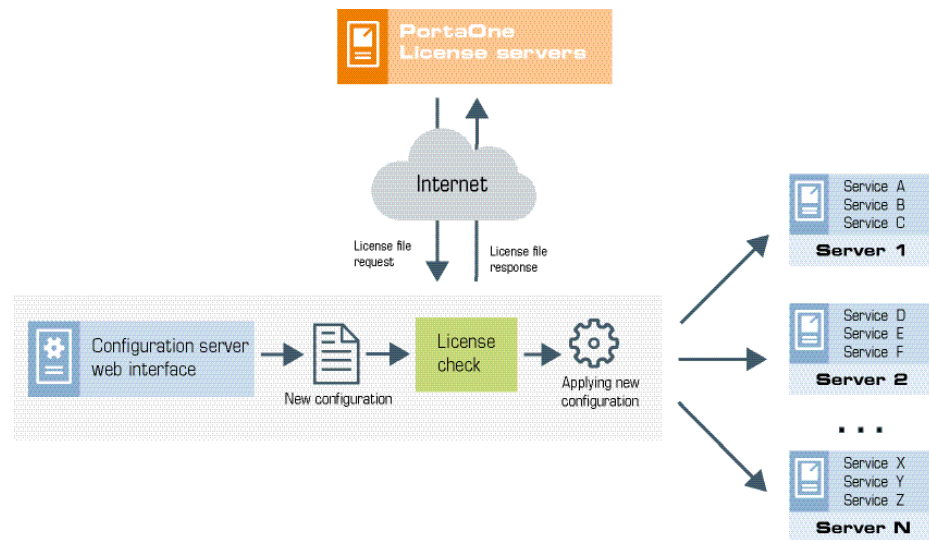
A local copy of the license file is stored on the Configuration server and then distributed to the remaining servers. Each individual application uses it to verify that this service can run as a part of this installation – a valid license file is necessary for any application to operate. The local copy 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, 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.
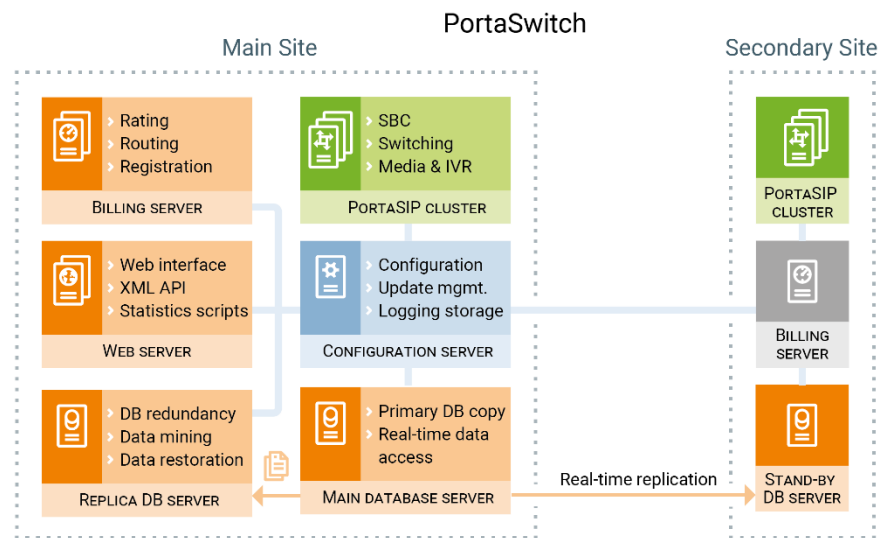
So if the secondary site detects that the main site has become unavailable, the "stand-alone" mode is activated on the secondary site and now it provides the service to end users using the latest available snapshot of the service configuration. The xDRs for consumed services and changes in balance are 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.
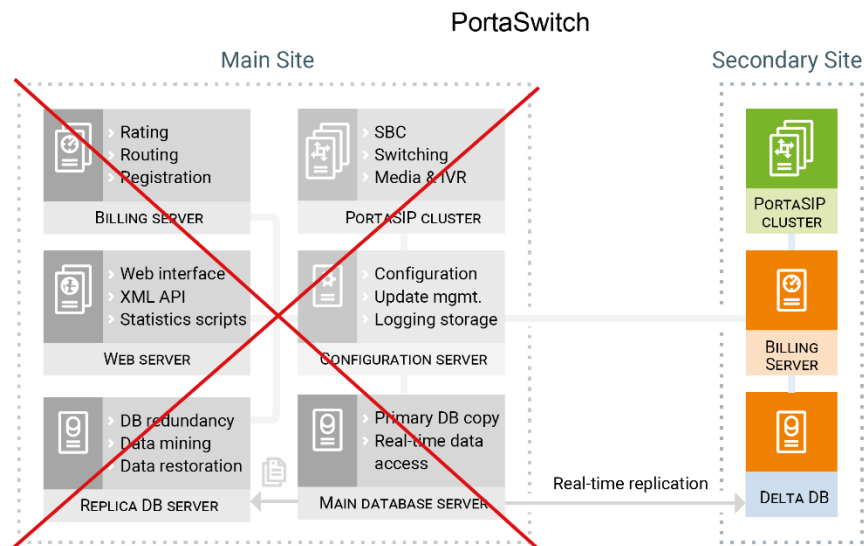
Once the main site becomes available again, the secondary site starts the process of synchronizing all of the accumulated changes to the main site and then the secondary site switches back to its normal ("stand-by") mode.
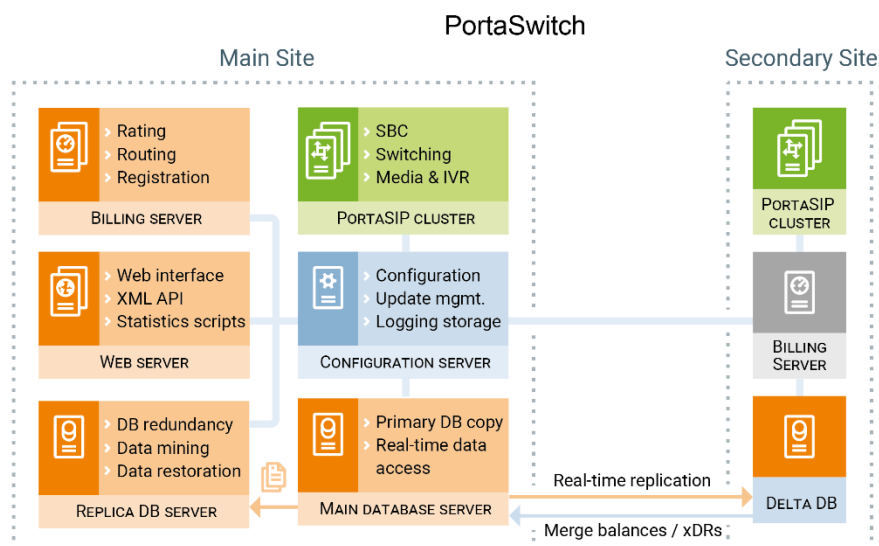
Typical deployment scenario

Let's consider the example of a possible PortaSwitch® deployment across multiple sites (a different deployment scenario might be fully cloud-

21

based). The "primary" site hosts a standard PortaSwitch® Clustered (the Configuration server, main and replica database servers, a cluster of PortaBilling® OCS and web servers and the PortaSIP® cluster).
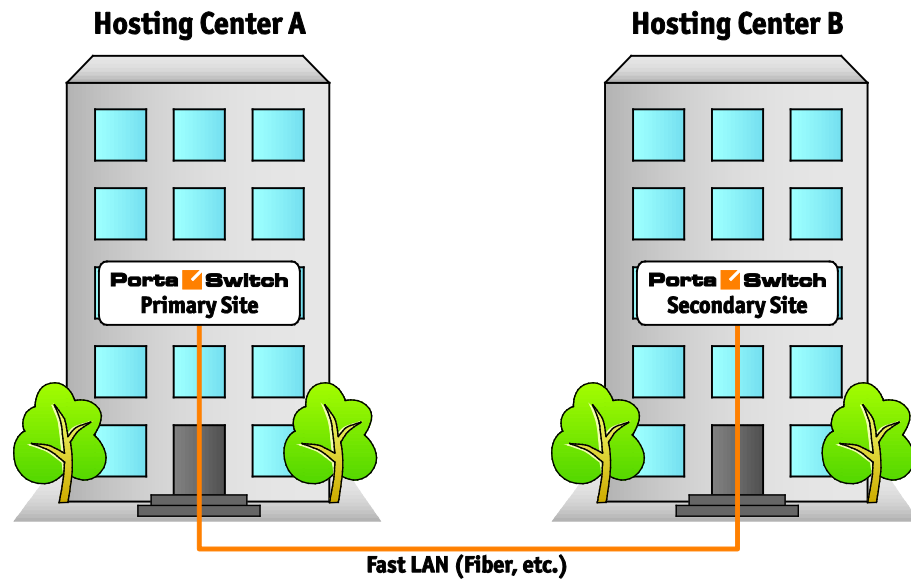
Within its "normal" mode of operation at the remote site:

- 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 OCS 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 OCS 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.
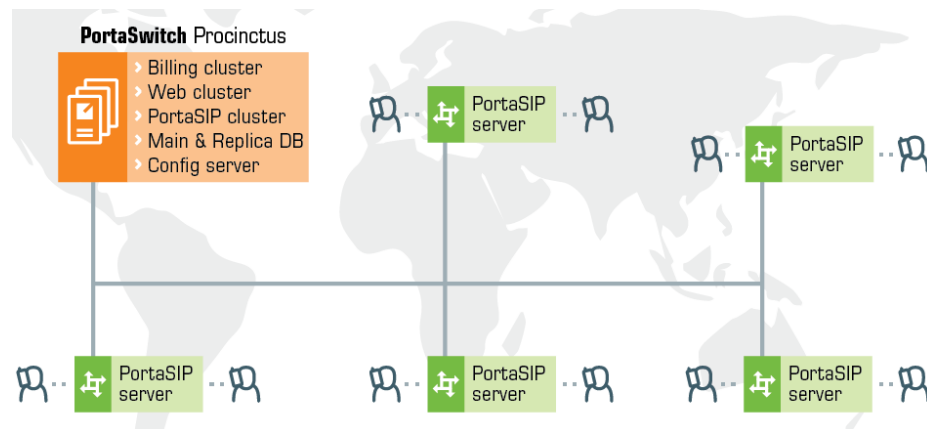
Whatever the choice, there is an essential requirement to provide proper interconnection between the sites. There are a lot of ways to organize the sites into a single corporate network; the selection of the technology depends on existing network infrastructure, equipment or capabilities of your network provider.

Regardless of the technology you choose, all PortaSwitch® servers must be connected via virtual (or physical) Layer 2 connection(s) and be configured as hosts in a single virtual (or physical) private network.

When disaster strikes

If there is an outage (for instance, a motherboard failure) on a single server (e.g. PortaBilling® OCS server #1) at the primary site, the primary site continues to operate as usual. Another server within the cluster (PortaBilling® OCS 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 OCS 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® OCS 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.

Promoting a secondary site to the main one

If at any point your main site is badly damaged (e.g. by fire or floodwaters) and is beyond repair, you can re-configure your secondary site to act as the main one and have a fully-functioning PortaSwitch®.

The procedure consists of the following steps:
- Initialize the Configuration server. On the secondary site, select the server that is most suitable to serve as the Configuration server and run the Configurator install script on it.
- Restore the Configuration server database from the backup. The Configuration server database backup is performed daily and is stored on the secondary site's database server(s). Use the `cfgdb.sh` script to restore the Configuration server database from its backup copy and run all the services required for its functioning (such as MySQL, Apache, etc.).
- Adjust the restored configuration.
  - On the Configuration server web interface, move servers from the failed main site to the secondary site (change their **Site name** property).
  - Make the standby database the master database: create a master database, replicate all the data from the standby database to the new master database and then delete the standby database.
  - Add instances that weren't initially configured on the secondary site (such as web servers, CDR importer, etc.).
- Apply the configuration.

From that point on, the secondary site acts as the main one and provides a fully-functioning PortaSwitch®, without restrictions or limitations.

PortaSwitch® application components
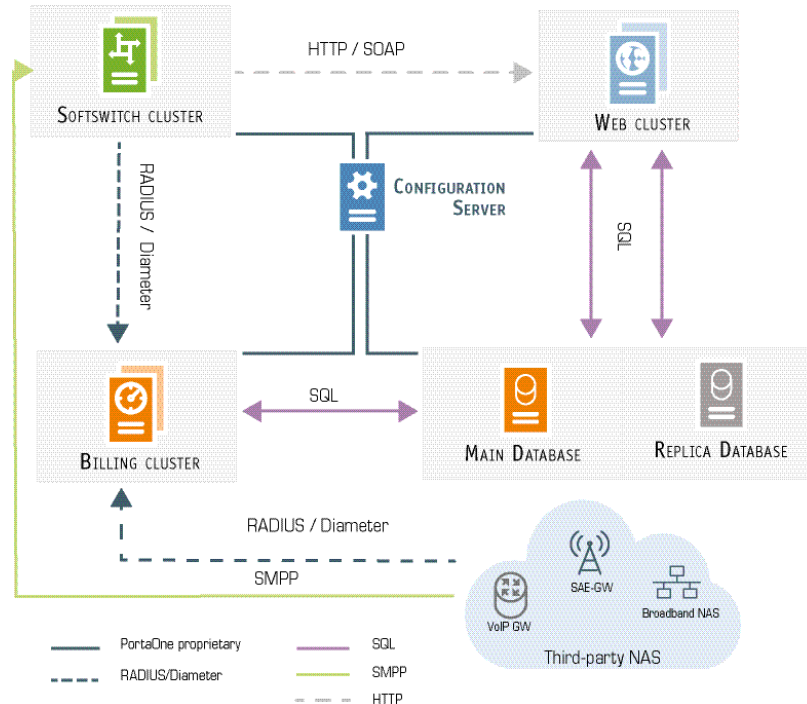
Every PortaSwitch® application (e.g. OCS server, web interface, etc.) consists of a combination of processes (subcomponents such as the RADIUS or Diameter daemons, the Apache server, statistics collection tools, etc.). All of these processes possess certain functions and interact with a number of other 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 OCS server

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

Protocols and ports used by PortaSIP®

PortaSIP® components listen on following ports:

Subcomponent	Protocol	Ports	Details
Dispatching node			
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
Processing Node			
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
B2BUA	UDP	5070	Base SIP port for B2BUA workers
B2BUA	TCP	5064	B2BUA telnet interface
B2BUA	TCP	5000	Call controller API
B2BUA	UDP	5059	B2BUA sibling communication
Registrar	TCP	5065	Registar transport ports
Registrar	UDP	5065	Registar transport ports
Subscription manager	TCP	5066	Subscription manager transport ports
Subscription manager	UDP	5066	Subscription manager transport ports
IMAP server	IMAP	143	IMAP transport
IMAP server	IMAPS	993	IMAP over SSL
Log Master	TCP	10000–10800	

Protocols and ports used by the Web server

Subcomponent	Interacts with	Protocol	Ports	Details
TaskStack	Main DB Server	TCP	3306/3307	MySQL Database server connection
	Replica DB Server	TCP	3306/3307	MySQL Database server connection (SELECT requests only)
Apache/FCGI	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)

Cassandra	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)** – 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.
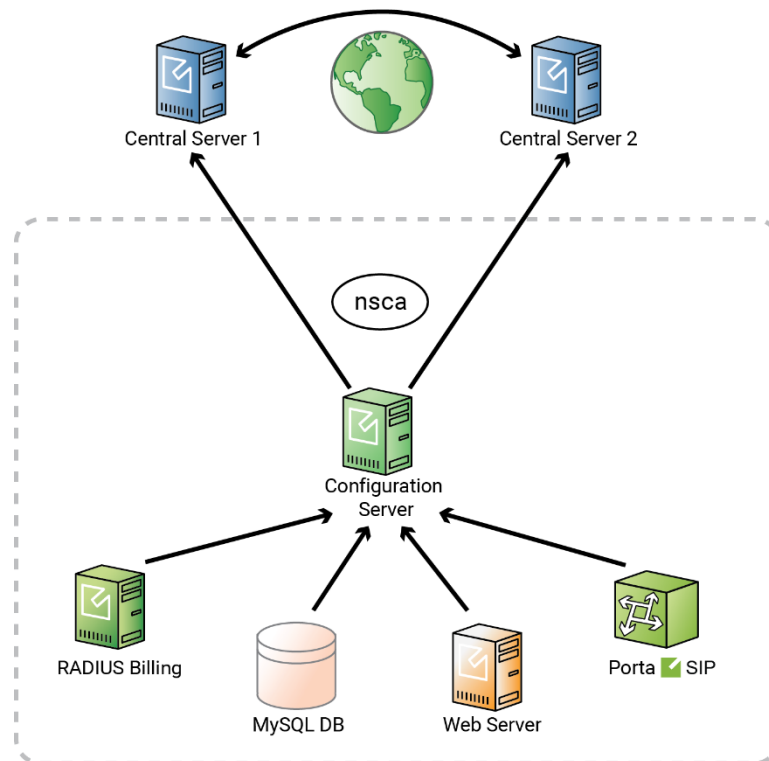
Figure 1-1 – PortaOne NSCA-based monitoring

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.
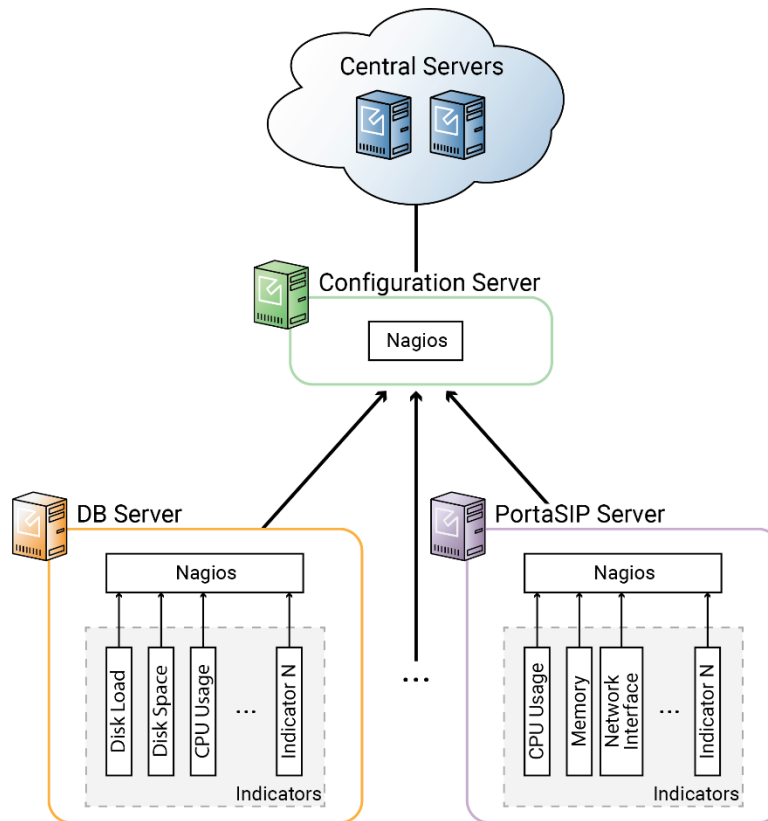
Figure 1-2 – Nagios monitoring scheme

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 addresses 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.
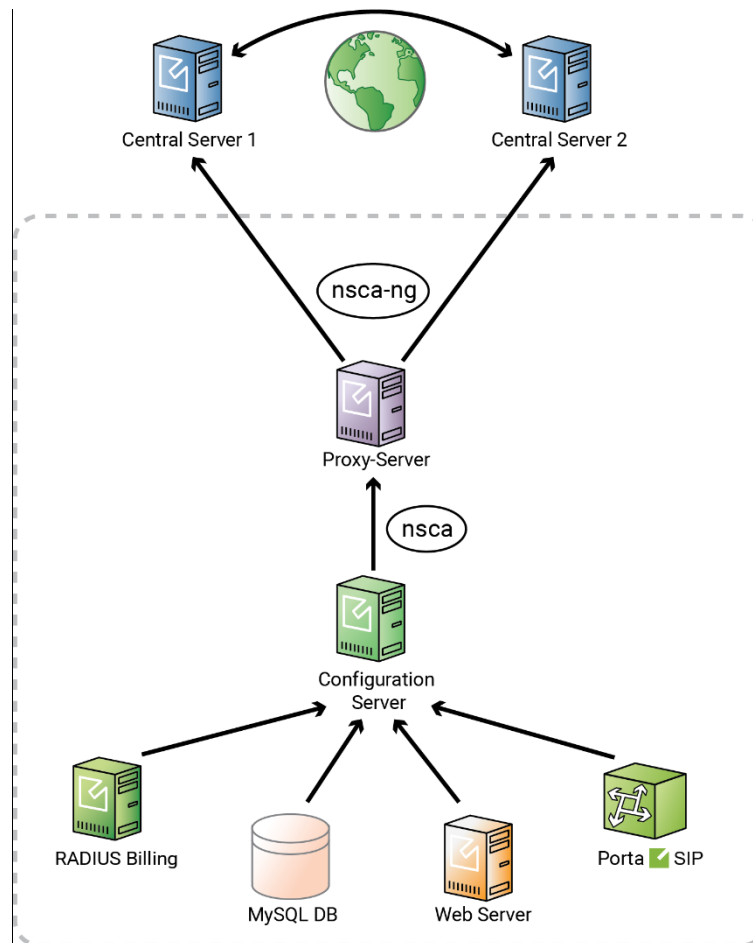
Figure 1-3 – PortaOne NSCA-NG-based monitoring

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 supplied with a PortaSwitch® installation with default settings. The PortaOne support team can adjust these 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. Before adjusting the settings, discuss your intended changes with PortaOne support so they can continue to respond to any emergency in your system in time.

The list of all indicators' names and definitions can be downloaded from **here**.

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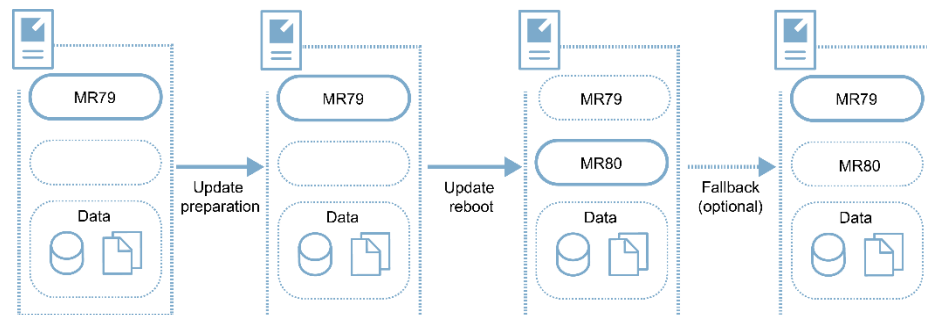
After some time when you wish to update to an even newer release, this partition is wiped clean and the new version of the code is loaded into the recently emptied location. Then the process described above repeats.

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 that may affect the system's performance while they are carried out. For example, adding a column to a table in MySQL prevents all other queries to that table from being executed until the operation is complete (another advantage offered by PortaBilling® with Oracle Real Application Cluster 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 number of blocking data structure modifications and the amount of time required to apply them. We do it by using the following techniques:
  - Zero downtime update technology. During the update of the main site, the secondary site processes the load.
  - PortaBillling® with Oracle Real Application Cluster allows Oracle online table redefinition to be used if there is no secondary site.
  - Dual Version. This is a PortaOne solution that allows gradually migrating the subscribers from an older release to a newer one.

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

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 must 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 – for which at least two sites are required. On the main site extend the capacity of your PortaSIP® to include dispatching SBC instances.

The upgrade is performed using the following procedure:
- First, the usual upgrade preparation is performed (verification of custom modifications, hardware check for compatibility with new OS, etc.).
- Then, secondary site B is switched to stand-alone mode. The dispatching SBC sends new registration and call initiation attempts to site B.
- Though PortaSIP® on main site A is switched to "maintenance" mode, active calls remain connected until they are completed.
- After all existing calls are completed on site A and there is no more live traffic there, the upgrade process begins on that site. A new version of code is installed on all servers, database changes are applied and servers are rebooted (excluding the dispatching SBC).
- The dispatching SBC cluster undergoes the update process sequentially: standby mode is enabled on the first dispatching SBC node. The virtual IP address is switched to the second node, which becomes active and handles all call and registration requests. Then the update process starts on the first node. Upon

completion, the update process starts on the second node. The second node is put into standby and the virtual IP address switches to the first node, which then becomes active. After the update on the second node completes, its standby mode is disabled.

- Throughout this time, customers are able to use services via site B as usual; calls are charged, balances are updated accordingly and xDRs are written. Find the list of limitations in the **Stand-alone mode restrictions** section.

- Finally, when the upgrade is completed on site A and all services there are verified to be working properly, site A becomes activated. 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 are directed to site A (using the same tools as described earlier). Any calls in progress on site B finish normally.

- 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 the changes in the database structure from site A, a new version of the software is installed and its servers are rebooted.

- When the update finishes, 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 replicated to site B so it has an up-to-date copy of all service configurations.
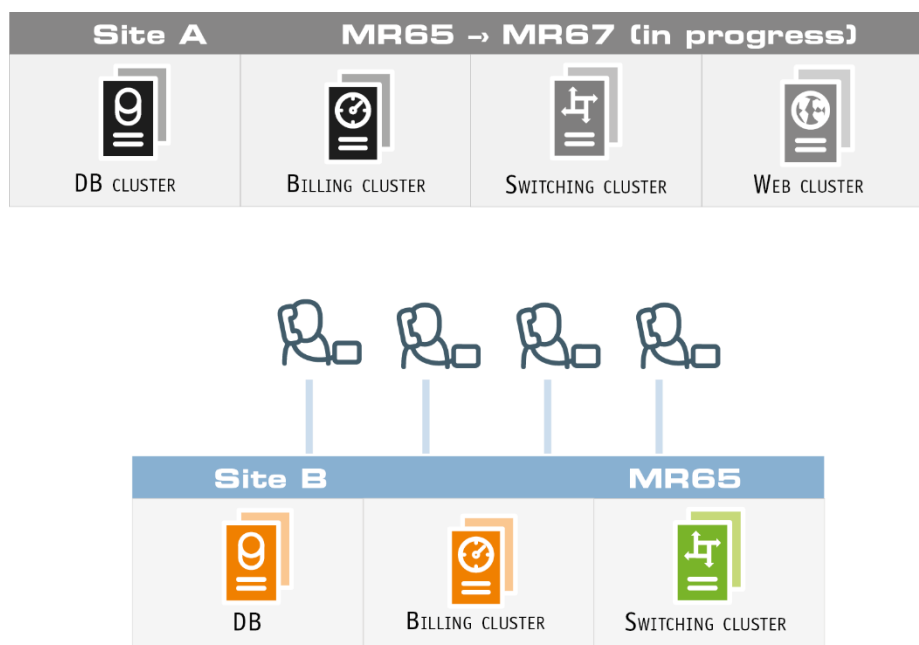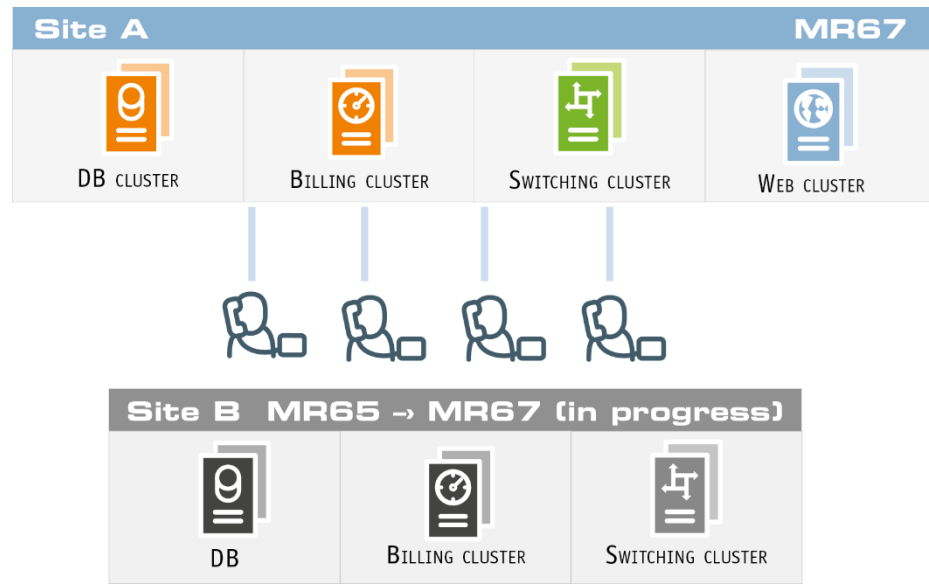
Custom modification management

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.
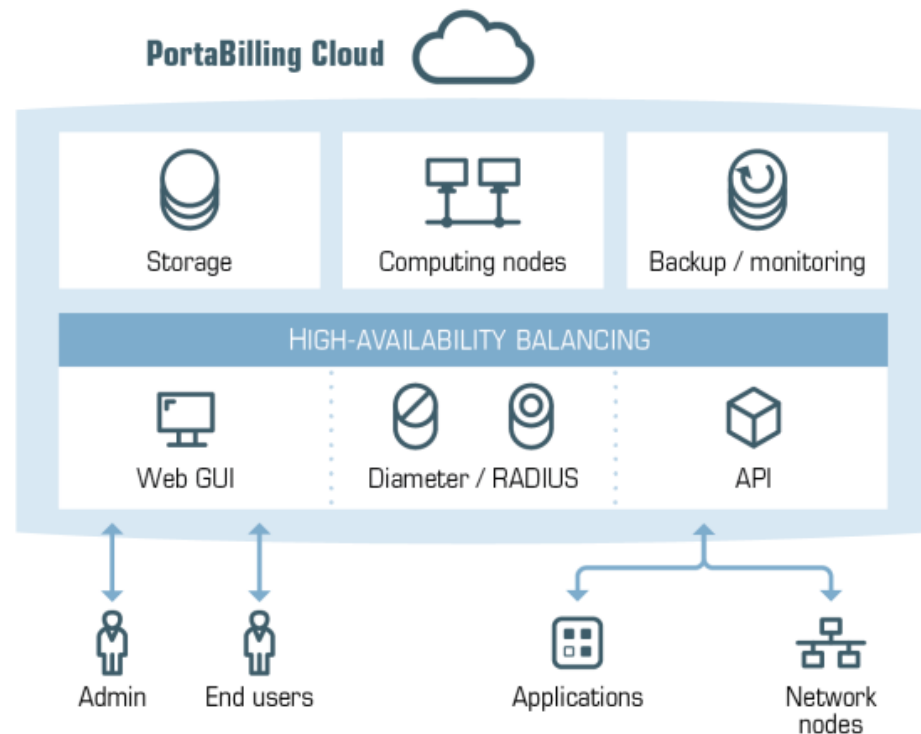
The key features of the cloud-based PortaSwitch® solution are:
- Near-zero CAPEX and reduced OPEX – By deploying PortaSwitch® in a cloud, you save on hardware purchases and provisioning. The cloud infrastructure is maintained by PortaOne, which drastically reduces the load on your administrative staff.
- 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 number of 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, please read the **Dual Version PortaSwitch®** chapter);
- 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.

Site-to-site VPN for PortaSwitch® deployed on the premises and in the cloud

There are several ways to deploy PortaSwitch®: on the premises on hardware servers, in a cloud or as a combination of both (e.g. you disperse PortaSwitch across multiple sites for redundancy or deploy a Dual Version PortaSwitch® for gradual customer migration).

For PortaSwitch® installations deployed both in a cloud and on the premises, a reliable and secure interconnection must be established between these two locations. In practice, this means building a VPN that combines networks from both the cloud infrastructure and the on-premises datacenter. Typically, site-to-site VPNs use tunnels to encapsulate data packets within normal IP packets and forward them over public IP-based networks, using encryption to ensure privacy.

The configuration of such a tunnel involves the following:
1. set up a VPN endpoint within a cloud network;
2. set up a VPN endpoint within an on-premises network;

3. make sure that both VPN endpoints freely exchange TCP/UDP traffic on ports 500 (ISAKMP) and 4500 (NAT traversal);
4. enable a tunnel between the two VPN endpoints and check the connectivity.

VPN endpoint in a cloud network

To set up a VPN endpoint in a PortaSwitch® cloud network, submit a request to the PortaOne Support team. The request must include the following information:

- a public IPv4 address for a VPN endpoint located on the premises;
- the IP address of the private on-premises network that will be added to the VPN.

Once a VPN endpoint in a PortaSwitch® cloud network is configured, you are provided with its public IPv4 address, pre-shared key and a private cloud network address.

VPN endpoint in the on-premises network

To set up a VPN endpoint in the on-premises network, either a VPN-capable device or software-based router is required (both must support standard-based IPSec encryption). Such a device/software router must be placed in the on-premises network and have a public IPv4 address assigned to it. It must also comply with the following ISAKMP policy options:

- ISAKMP protocol, version 1;
- exchange type: main mode;
- authentication method: pre-shared keys;
- encryption algorithms: AES-128-cbc, AES-192-cbc, AES-256-cbc;
- authentication algorithm: SHA1 (also called SHA or SHA1-96), SHA-256, SHA-384;
- Diffie-Hellman group: group 1, group 2, group 5; and
- IKE session key lifetime: 28800 seconds (8 hours).

Additionally, support of the following IPSec policy options is required:

- IPSec protocol: ESP, tunnel-mode;
- encryption: AES-128-cbc, AES-192-cbc, AES-256-cbc;
- authentication algorithm: HMAC-SHA1-96;
- IPSec session key lifetime: 3600 seconds (1 hour); and
- Perfect Forward Secrecy (PFS): enabled, group 5.

Please refer to **https://docs.cloud.oracle.com/iaas/Content/Network/Tasks/configuringCPE.htm** for useful details about a particular VPN endpoint configuration.

Enable a tunnel between VPN endpoints

Provision the public IPv4 address, pre-shared key and private cloud network address of the VPN endpoint configured in the cloud onto an on-premises device (or software router) so that a tunnel between the two VPN endpoints can be established.

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.
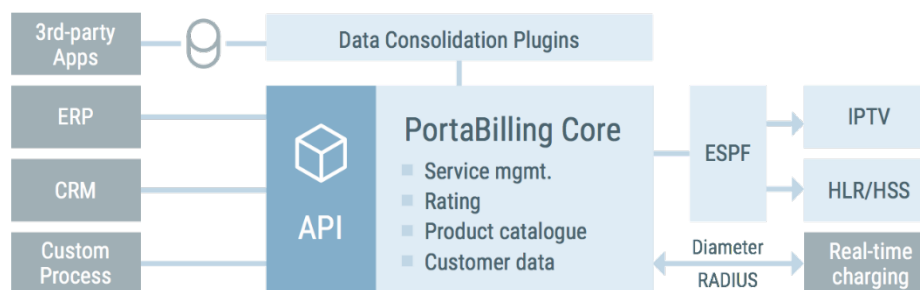
API for data operations

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).
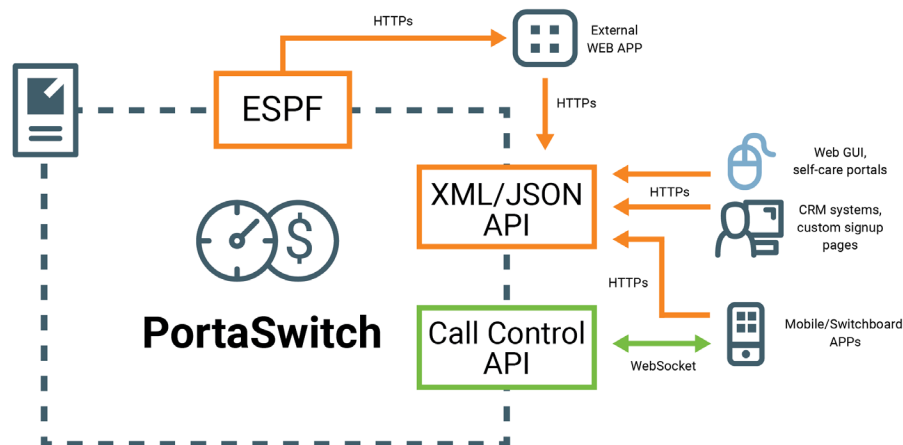
Performing operations via API has several advantages:

- 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**.

Extended data search via the API

The `get_customer_list` API method allows filtering customer data by using attributes that are relevant to them (e.g. by customer's home city). However, this might not be sufficient to perform advanced customer and account searches, e.g. to search for US-based customers who have spent over $100 on voice calls in a previous billing period.

The `get_extended_data_list` API method enables you to do just that. You construct the API request and define your own search criteria as input parameters. For example, you can search for customers who live in Toronto and use the EasyCall and SuperCall products.

The API method retrieves the desired list and delivers consolidated information about customers to your CRM application. Thus, for the example above, it includes information about customers, their accounts and their products. This allows you to use PortaBilling® data via your CRM applications to execute marketing campaigns, formulate reports, etc.

The `get_extended_data_list` method is only available in JSON format. Since you create your own API requests, deeper knowledge of the PortaBilling® API is required.

The `get_extended_data_list` API method enables you to filter customers and accounts by using the attributes of such entities as:

- customers,
- accounts,
- products (both main ones and add-ons),
- customer classes and
- invoices.

Upon your request, this method can be expanded to operate using other PortaBilling® entities.

Thus, having the ability to perform an advanced data search via the API gives you the following benefits:

- The ability to retrieve data from PortaBilling® directly from your CRM or other external apps;
- Secure access to the PortaBilling® database; and
- Flexibility in forming search queries.

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.

Provisioning data to external systems

To automatically update your user's configuration in external systems (e.g. mobile core network components, IPTV platform, etc.) from PortaBilling®, use the External Systems Provisioning Framework (ESPF).

The ESPF captures the changes in a user's configuration in PortaBilling® and then notifies the external systems about them.

To make this happen, specific changes in a customer and/or account configuration are recorded as **events** in PortaBilling®. The ESPF monitors these events, processes them and sends the updated events via HTTP protocol to your custom web application.

The application receives the request, retrieves the necessary data (e.g. MSISDN, IMSI) from PortaBilling® via the API and triggers the API call in the external system for some action (e.g. activate a SIM card in the HSS or create a new subscriber in the IPTV platform.)
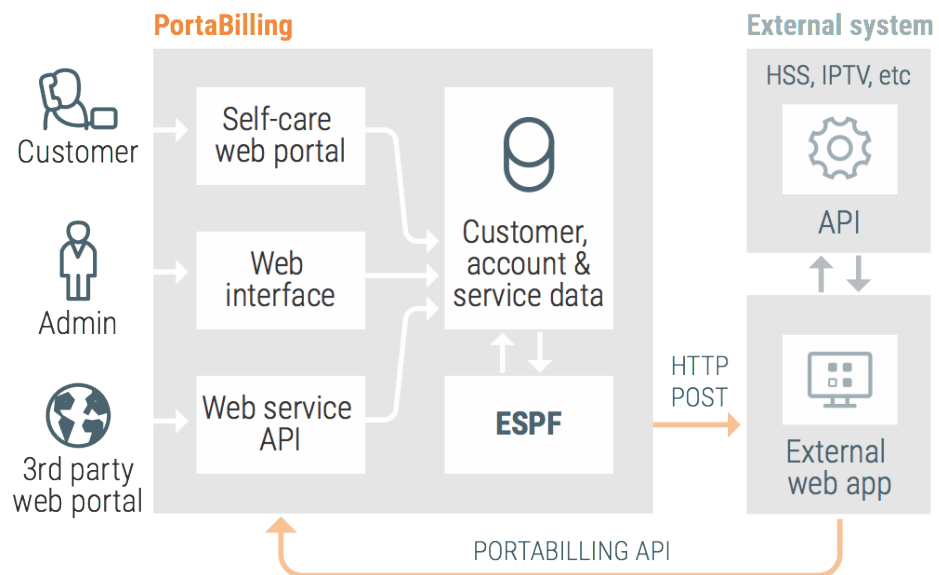
Delivering data via HTTP enables you to use any programming language for your web application and run your application on any of your web servers (e.g. on premises or hosted in a cloud, etc.).

Another approach to data provisioning is the use of dedicated event handlers. An event handler is a special Perl module in PortaBilling® that provisions a particular external system using this system's API. There is a separate handler for each external system that PortaBilling® is integrated with. Find the list of available event handers in the **External Systems Interfaces Guide.**

If you need to integrate a new external system and provision data to it, you can either create your own event handler or request development from PortaOne.

The ESPF simplifies the integration with the external systems that you deploy in your network infrastructure. PortaBilling® provides a single location for data provisioning management.

Data consolidation plug-ins

Let's say you use PortaBilling® for customer management and a NMS (Network Management System) for monitoring network performance, circuit management and configuration, etc. When checking customer information (e.g. during troubleshooting) you must login to both PortaBilling® *and* to the NMS.

Instead of managing two systems separately, you can integrate the NMS with PortaBilling®. Create an integration plug-in, install it in PortaBilling® and operate with the NMS from the PortaBilling® web interface.

Each time you open customer data, PortaBilling®:
- dynamically retrieves the network information related to the customer from the NMS via the API, and
- displays the consolidated data on the web interface.

The NMS data is then displayed on the custom panel.

When you modify the information (e.g. change bandwidth limits or reconfigure a router), the data is automatically sent to the NMS.

An integration plug-in is a set of application files that contain the logic for a new custom panel (e.g. fields, types, default values). While installing the plug-in, specify whether the panel appears in the main navigation menu or at a specific entity level (e.g. customer, reseller). Upon installing the plug-in, the custom panel for system operation appears in PortaBilling®. This way, you can integrate PortaBilling® with any external system that supports the REST API.

Consider the following example:

Asgard Telecom has a contract management system (CMS) to store and manage customer contracts. They require:
- real-time interaction between the CMS and PortaBilling®, and
- the ability to manage the contracts (e.g. Contracts feature) via the PortaBilling® web interface.

Asgard Telecom implements a plug-in for the Contracts feature. After the plug-in installation, a new Contracts panel is available at the customer level. It permits the customer contract details to be managed. When an administrator saves a contract, the plug-in pushes the data to the CMS via the API. Each time an administrator opens the Contracts panel, the data is retrieved from the CMS.

This solution allows you to integrate PortaBilling® with the external systems and manage all of the data from a single place. Aside from improving workflow, it gives you additional flexibility with introducing individual business solutions using PortaBilling®.

Please contact the PortaOne® support team for assistance with how to install integration plug-ins.

3. PortaOne iPaaS

PortaOne iPaaS

PortaOne iPaaS (Integration Platform as a Service) is a provisioning solution that enables you to deploy new functionalities (e.g. provision new CPE profiles or add new payment processors) through containers that run in the cloud. This eliminates the need to update all of PortaSwitch® to the newest release. Thus, you reduce the time to market for new services.

A container is a portable software unit where all the components of new functionality (code, runtime, system tools, system libraries and settings) are stored as one package. Containers are deployed in the cloud-based PortaOne iPaaS independently from the PortaSwitch® installation. As a result they are easily modified and extended.
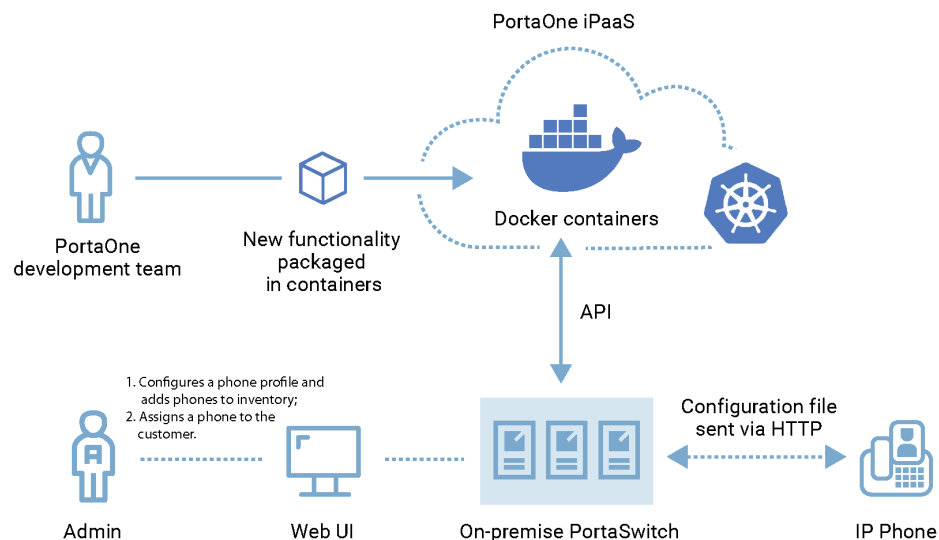
To start using new features via the PortaOne iPaaS, subscribe to it. To find out about our subscription terms, contact our **Sales department**.

Benefits:

Service provisioning via PortaOne iPaaS:

- Significantly saves your time and resources since you do not need to update your installation to receive a new functionality.
- Reduces time to market for new functionalities, making you more competitive.
- Improves the customer experience. You can provide your customers with the newest devices and functionalities within a short period of time.

CPE provisioning via PortaOne iPaaS

The first device that's available for provisioning via PortaOne iPaaS is Yealink W80B – an IP multi-cell base station. To read about other IP phones provisioning, refer to the *Customer Premises Equipment (CPE) provisioning and inventory* chapter in the **PortaBilling Administrator Guide**.

This is how CPE provisioning via PortaOne iPaaS works:

Let's say the service provider, Easy Call, wants to provision Yealink W80B to their customers. Since this device model is available via PortaOne iPaaS, Easy Call subscribes to the CPE provisioning service and enables it on their installation.

The rest of the configuration process for creating a new CPE profile and managing the IP phone inventory in PortaBilling® remains the same as before. Refer to the **Auto-provisioning of IP Phones** handbook for the detailed configuration.

When an administrator creates a CPE profile for Yealink W80B, PortaBilling® queries the CPE provisioning container via the API to generate a CPE profile. The CPE provisioning container returns the generated CPE profile and it is stored in the PortaBilling® database along with the internal CPE profiles.
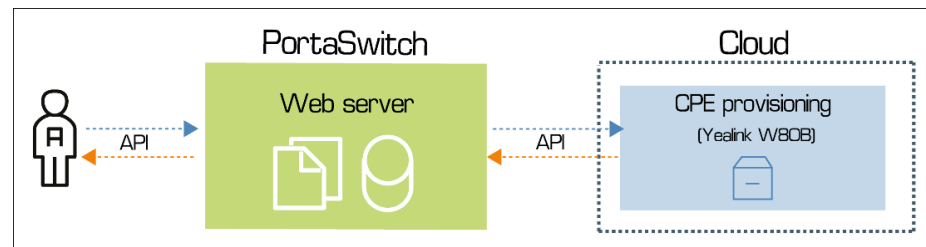
Then the administrator creates the Yealink W80B CPE device and assigns it to an account. PortaBilling® generates a configuration file with the CPE device parameters (username, password, device's MAC address, etc.) and sends it to the CPE provisioning container. This configuration file is used in the CPE provisioning container to adjust Yealink W80B's internal configuration.

Now Yealink W80B is ready to use by end users.

Configuration:

Once you subscribe to PortaOne iPaaS, PortaOne support provides you with credentials to enable CPE provisioning on the Configuration server.

- Go to the **Web_Cluster.Global.ExternalCpeProvisioningService** group

and specify the **CustomerName** and **InstallationName** options as defined in RT.
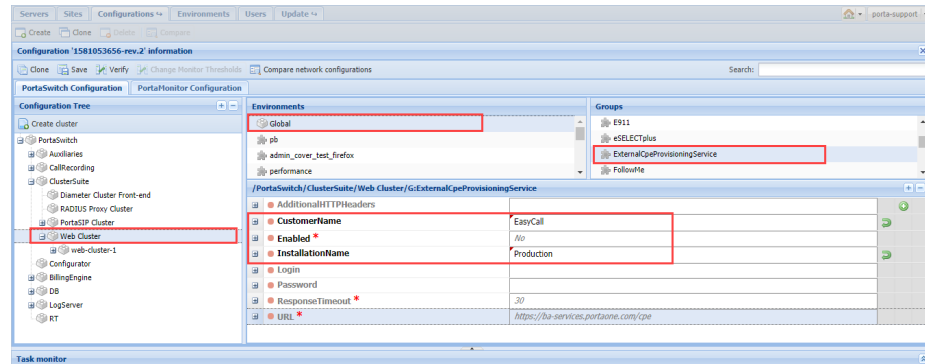
- Go to a specific environment and set the **ExternalCpeProvisioningService.Enabled** option to **Yes**.
- Specify the credentials provided by our support team under the **Login** and **Password** options.
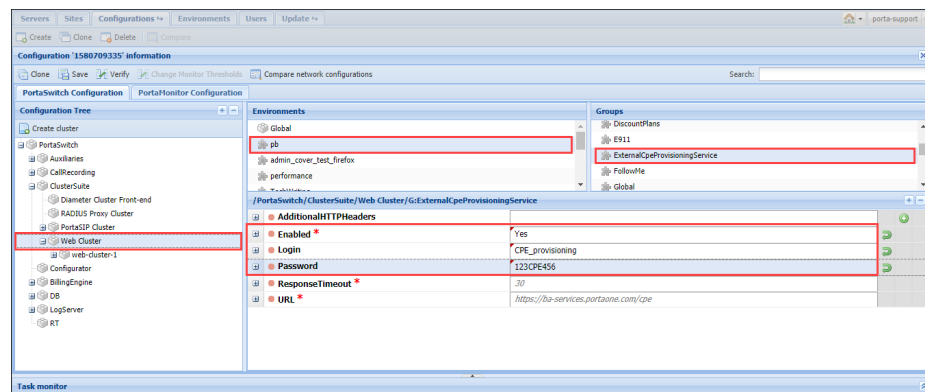
Payment processor integrations via PortaOne iPaaS (Stripe payment processor)

The first payment processor that is currently available in PortaBilling® via PortaOne® iPaaS is **Stripe**. Stripe is one of the most widely used payment processing systems in the world and supports over 135 currencies. It also supports token-based payments, enables you to accept online payments right away, and allows your customers to set up recurring auto-payments.

PortaOne® iPaaS subscribers running MR87 and above can select and use Stripe from within the PortaBilling® web interface and receive credit card payments with minimal integration efforts. For MR55–MR86, the integration process with Stripe is also available via a backport patch for the specific MR.

In this integration, PortaOne® iPaaS communicates with the on-premise PortaSwitch® and the Stripe payment processor transfers the payment data via API to the PortaOne® iPaaS, which acts as a cloud intermediary.

Let's see how Stripe payment processor can be integrated in two different scenarios:

Scenario 1. Stripe integration for PortaSwitch® MR86 and earlier

Panda Telecom runs MR70 and requires support for online payments using Stripe. Panda Telecom turns to PortaOne® to request integration and pays the PortaOne® iPaaS subscription fee. The rest of the process is as follows:

- The PortaOne® development team prepares a backport patch for the exact PortaBilling® version used by Panda Telecom (for an additional price).
- The backport patch for Stripe integration is applied within Panda Telecom's PortaSwitch® infrastructure.
- Panda Telecom's PortaSwitch® authorization credentials for iPaaS are set on the Configuration server.

As a result, Stripe appears on the list of payment processors available for Panda Telecom.

Next, Panda Telecom signs a contract with Stripe for a merchant account and receives the Stripe API key. After configuring the Stripe merchant account from within the PortaBilling® web interface, Panda Telecom can use Stripe as a payment processor. The whole integration process takes 1-2 weeks.

Scenario 2. Stripe integration for PortaSwitch® MR87 and later versions

StarlightNet runs MR87 (or later versions) and wants support for online payments with Stripe. In this case, the process is similar but faster (up to 1 week), as there is no need for a backport patch:

- StarlightNet turns to PortaOne® to request integration and pays the PortaOne® iPaaS subscription fee.
- StarlightNet's PortaSwitch® authorization credentials for iPaaS are set on the Configuration server.

As a result, Stripe now appears on the list of payment processors available for StarlightNet. Next, StarlightNet signs a contract with Stripe for a merchant account and receives the Stripe API key. After configuring the Stripe merchant account within the PortaBilling® web interface, StarlightNet can use Stripe as a payment processor.

If you would like to add any other payment processors besides Stripe, this can also be done with the help of PortaOne® iPaaS. Such new integrations take up to several weeks and do not require an update.

For the full list of currently available payment processors that support token-based payments in PortaBilling®, refer to *Appendix F* of the **PortaBilling® Administrator Guide**.

Configuration:

The initial configuration for enabling Stripe is done via the Configurator by PortaOne® admins after you subscribe to PortaOne® iPaaS. Next, you need to create a merchant account with Stripe (might take a while until you get verified) and get your API key. Using this API key, you can configure Stripe as a payment processor within PortaBilling® using the step-by-step instruction in the **Configuring Stripe payment processor handbook**.

4. Dual Version PortaSwitch®

Concept of Dual Version PortaSwitch®

Dual Version PortaSwitch® is a solution for gradual migration to a newer software system version across several MRs using a two-system setup and customer batching. It allows service providers to gain access to the new features, secure their existing platform's overall stability during an update, and carry out the update at a comfortable pace.

Dual Version PortaSwitch® consists of two systems – source and target – that operate simultaneously.

- The **source system** runs the current release (e.g., MR65).
- The **target system** initially mirrors the source system and is then updated to a new release (i.e., MR65→MR80).

To eliminate human errors and ensure stable and correct processes of service configuration and data transfer, Dual Version PortaSwitch® provides technical solutions that help migrate some of the existing customers to the new release. Customer migration is done in batches. Immediately after the migration of a given customer batch, the services for it can be run with minimum reconfiguration efforts.

The available technical solutions in Dual Version PortaSwitch® address:

- Customer data transfer.
- Transparent routing of the calls to the system serving the customer.
- Proxying RADIUS/Diameter requests to the system serving the customer.
- Common API entry point.
- Distribution of number porting requests across systems based on the customer record location

About the target system setup

The target system starts as a full database copy of the source (production) system (e.g., MR72). The database snapshot of the production system is copied to the target system, where it is upgraded to the new software version (e.g., MR85). As a result, all configuration entities such as products, tariffs, and rates are cloned and ready-to-use on the target system.

The dedicated provisioning scenario configures interconnection between both the source and target systems, installs all required code modifications, and enables all components required to run Dual Version PortaSwitch®, such as ESPF handlers, forwarding registry, etc.

Currently, the updates that "jump" over several release versions require obligatory checkpoints every 15 releases (currently MR55-6→MR70-6 or MR70-6→MR85-3).

Customer data transfer

Using Porter functionality, you can transfer some of your existing customers to the target system and immediately run services there with minimum reconfiguration efforts.
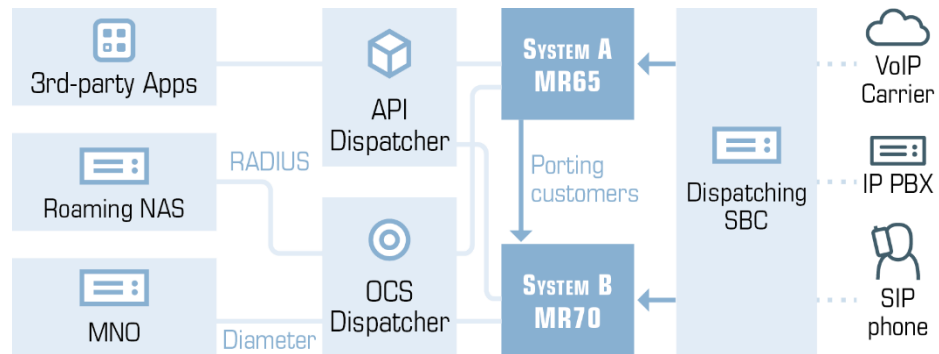
Porter is a script with which you export a customer's service and billing configurations and all of the dependent entities (accounts, products, service features, xDRs, invoices, etc.) from a previous "source" system and import them into the target.

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	Customers' configuration: • Customers and their accounts. • Customer and account service features. • xDRs. • Invoices. • Payment methods. • Customer sites. • IP Centrex elements – extensions, huntgroups and call queues. • Customer credit limit and payment history. • CPE and SIM inventory records. Generic configuration: • Custom access levels. • Customer classes.

	• Customer products and product groups. • Customer tariffs and rates. • Spending plans. • Subscriptions. • Volume discount plans and counters. • Dialing rules. • Fraud traffic profiles. • Custom reports • Call records.
Vendor	• Vendors and connections. • Vendor DID batches. • Vendor tariffs and rates. • Vendor xDRs

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 target 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 customers 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;
- Destinations;
- Destination groups and destination group sets;
- Nodes.

Porter operates in interactive mode. This means that if during a data transfer some entity appears to already exist in the production 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. 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.

For the data transfer to take place, the following entities must be pre-configured in the target system:
- A PortaBilling® root user;
- Users for your staff members;
- Currencies;
- Payment systems;
- Templates (invoice, tariff upload/download);
- Taxation methods;
- 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.
- Voice applications.

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.

Call records synchronization

Dual Version PortaSwitch® provides a mechanism for automatic synchronization of call records between source and target systems. As a result, you can access and download the call records in the Dual Version PortaSwitch® system regardless of the system (source or target) on which they were initially created.

Dual Version PortaSwitch® maintains the service integrity for your customers through constant synchronization of their call records with the help of a script that runs on the target system on set intervals of time.

When a new call record appears in the source system, the file is synchronized with the target system with the help of the script, even if the corresponding customer wasn't migrated to the target system yet. When the customer finally fully migrates to the target system, all call records will already be present there.

Parallel processing mode

Porter can migrate several customers in parallel, which allows you to increase data migration performance.

To enable parallel processing for Porter, the number of threads for data migration must be defined in the Porter configuration. This maximum thread number depends on the server's processing capacity and the load caused by other operations running on that same server. Contact PortaOne Support to evaluate the number of threads available for Porter in your installation.

Reverse transfer of customer xDRs

Customers may face issues with the target system after the transfer. If these issues should require deep investigation, an administrator can restore the customer on the source system.

To do this, the administrator runs the Retrop tool on the target system where the script performs a reverse transfer of customer xDRs as follows:
- first it analyzes the customer xDRs on both systems to find which ones of them can be returned (i.e. for services and entities that exist in both systems),
- it marks the customer record as Exported in the target system,
- then it imports the xDRs to the source system, and finally,
- it lifts the Exported status from the customer record in the source system,

65

- adjusts the customer's balance so that it corresponds to the customer's balance on the target system. If the customer has debit accounts their balance is also adjusted.

If the customer has used the services on the target system and therefore has new xDRs, Retrop generates the report for the administrator. The administrator then uses that to adjust the customer balance manually.

Note that a reverse transfer of customer xDRs must be treated as an emergency measure due to the following limitations:
1. The xDRs must be returned before the customer's billing period closes to avoid double charging, taxing and invoicing.
2. When restoring the customer record on the source system, the service configuration will be rolled back. This means that any configuration changes done on the target system (e.g. new service enabled) will be lost.
3. If the administrator has terminated the customer in the source system, the Retrop tool will not be able to perform a reverse xDR transfer.
4. If the administrator terminates a customer's account in the target system but the account still exists in the source one, the account will remain active after the Retrop tool performs a reverse xDR transfer and lifts the Exported status from the customer in the source system.

Cleanup of exported customer data

If you migrated some customers to the target system for testing purposes and later wish to return them to the source system, you need to remove the exported customer data.

To do this, use the Cleaner tool. Cleaner is a script that collects customer data and all its dependent entities and then deletes their records from the database. You can run Cleaner from either system in Dual Version PortaSwitch®, however, you must define beforehand where the data is to be removed from.

Thus, with Cleaner you can remove the following entities:
- Customers,
- Distributors,
- Products,
- Resellers,
- Tariffs,
- Vendors,
- Access levels,
- DID numbers,
- SIM cards, and
- CPE.

Similar to Porter, Cleaner allows you to delete entities one by one or several at a time. This ability to delete unnecessary data facilitates troubleshooting and optimizes the overall management of Dual Version PortaSwitch®.

Cleaner is designed to remove only data that has been transferred. Therefore, do not use it for simple system cleaning.

Call delivery in Dual Version PortaSwitch®

For the successful operation of Dual Version PortaSwitch®, it is essential that customers do not notice being moved from one system to another. That is, if John Doe is already in the target 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 take a closer look at how PortaSwitch® processes calls for John: first, when his record is still provisioned in source system A and then, when it is moved to target system B. To make and receive calls, the dispatching SBC's IP address is used to register John's phone.

Source system – outgoing calls

Let's say John calls his friend Harry in the UK.
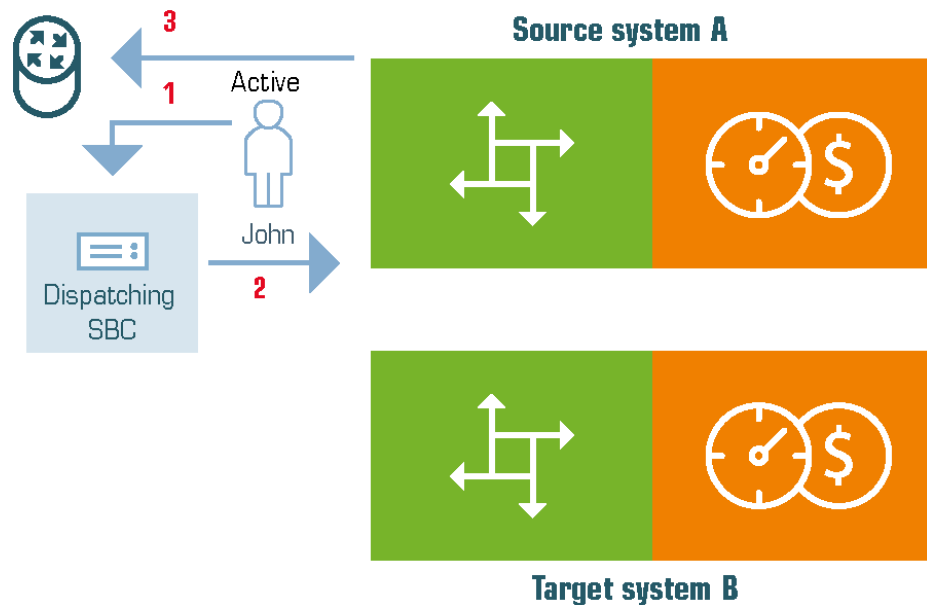
1. John's phone sends an INVITE request to the dispatching SBC (**1**).
2. The dispatching SBC searches for John's account in the database and detects that it belongs to system A.
3. The dispatching SBC routes the INVITE request to PortaSIP® on system A (**2**).
4. Within system A, PortaSIP® sends the authorization request to 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.

Source system – incoming calls

Consider this example when John receives a call from abroad.
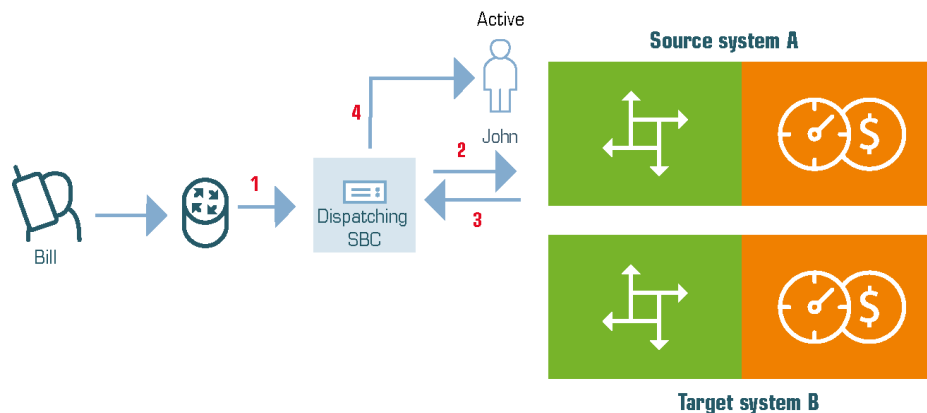
1. The vendor receives the call from the PSTN network and sends it to the dispatching SBC (**1**).
2. The dispatching SBC searches the database for the DID number that matches the destination number.
3. The dispatching SBC detects that the destination number corresponds to John's account and that it belongs to system A.
4. The dispatching SBC routes the call to system A's PortaSIP® (**2**).
5. PortaSIP® sends an authorization request to the respective PortaBilling®.
6. PortaBilling® authorizes the call and detects that the destination number matches the local account so it returns the result to PortaSIP®.
7. PortaSIP® checks that the account's IP phone is registered via the dispatching SBC, and therefore, it sends the call to the dispatching SBC (**3**).
8. When the dispatching SBC delivers the call to John's phone (**4**) it starts ringing.
9. When the call ends, PortaSIP® in system A sends accounting information to PortaBilling® to calculate the charges and create an xDR for John.

Target system – outgoing calls

The administrator has transferred John's record to the target system.

So now when John makes another call to his friend Harry in the UK, the following occurs:
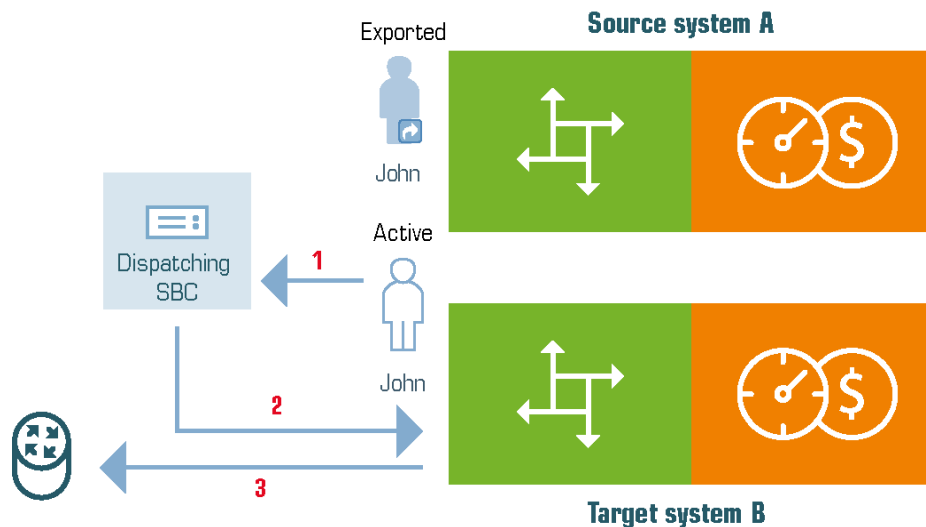
1. The call request from John's phone arrives to the dispatching SBC (**1**).
2. The dispatching SBC searches John's account in the database and detects that it is provisioned in system B.
3. The dispatching SBC routes the call to system B's PortaSIP® (**2**).

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 and establishes the call with Harry (**3**).
7. When the call ends, PortaSIP® sends accounting information to PortaBilling® to calculate the charges and produce xDRs.

Target system – incoming calls

When Jane dials John's number, the following occurs:
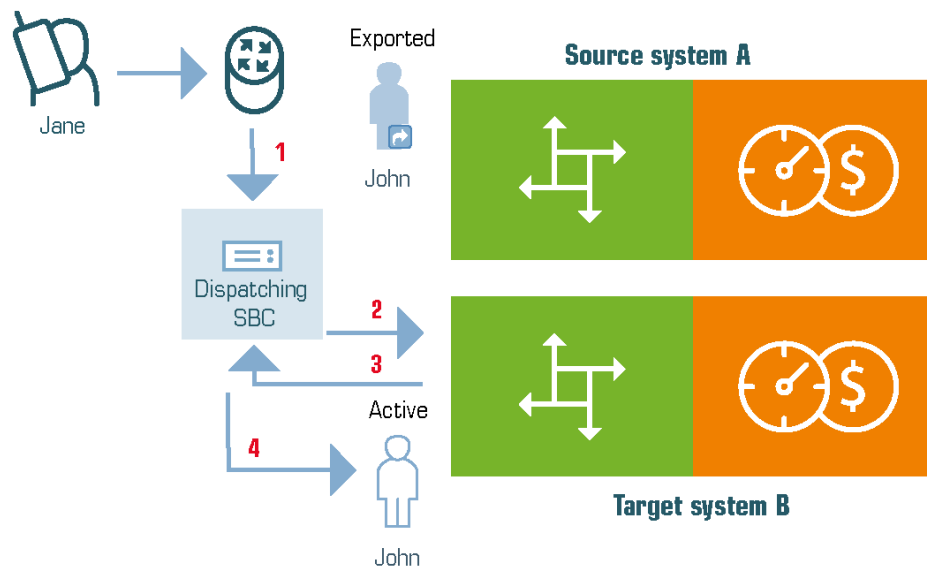
1. The vendor receives the call from the PSTN network and sends it to the dispatching SBC (**1**).
2. The dispatching SBC searches the database for the DID number that matches the destination number.
3. The dispatching SBC determines that the destination number corresponds to John's account and is provisioned 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 and notifies PortaSIP® that the call is intended for the local account.
7. PortaSIP® checks that John's IP phone is registered via the dispatching SBC, and therefore sends the call to the dispatching SBC (**3**).
8. The dispatching SBC delivers the call to John's phone and John and Jane begin their conversation (**4**).

9. Once the call ends, 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.
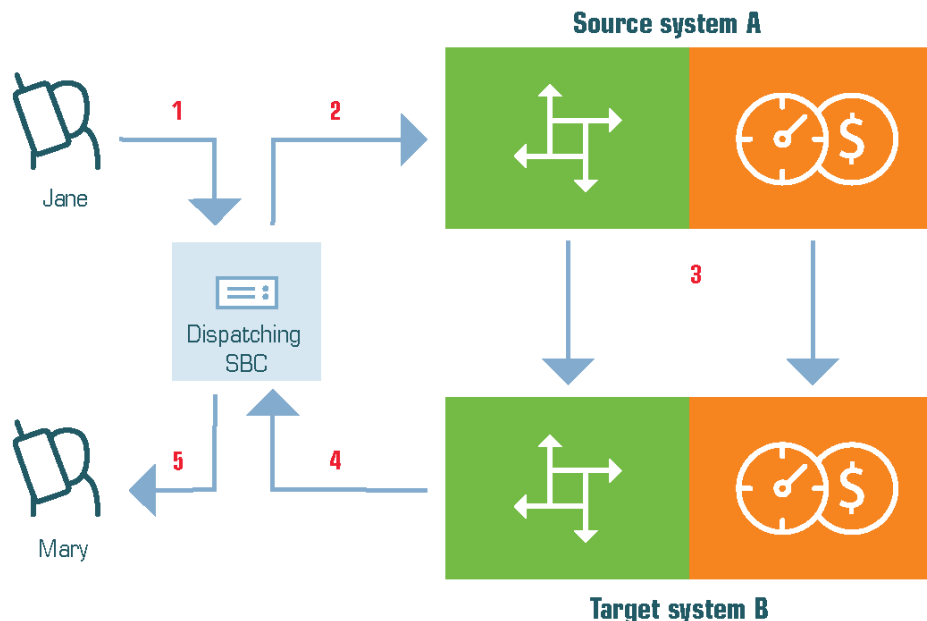
Thus, if Jane, from source system A, calls Mary, who is already in the target system B, the following occurs:

- The call from Jane's phone arrives to the dispatching SBC (**1**)
- The dispatching SBC sends a call request to PortaSIP® at the source system A for processing (**2**).
- PortaSIP® in system A authorizes the call in PortaBilling®.
- PortaBilling® detects that the destination number belongs to the account that has been moved to the target system and therefore instructs PortaSIP® to send the call to the system B.

- PortaSIP® on the system A routes the call to PortaSIP® on the system B, bypassing the dispatching SBC (**3**).
- Upon call authorization in the system B'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 PortaBilling® in system A creates an xDR for Jane's account while PortaBilling® in system B creates an xDR for Mary's account.

Fallback call delivery

To find which system to route an incoming call to in Dual Version PortaSwitch®, the dispatching SBC matches first the destination number and then the source number against the account/DID list in the database.

If the dispatching SBC cannot find the target system for some reason (e.g. there is no record in the database to match the destination number), it sends the call to a fallback system. In that system, PortaSIP® authorizes the call in PortaBilling® and receives the correct routing to deliver the call to the recipient. The target system is the fallback system by default.

You can configure a fallback system for the dispatching SBC to send calls. For example, if you have just deployed Dual Version PortaSwitch® and haven't yet migrated customers to the target system, define the source system as the fallback one. Select a billing environment from the source system in the **ForwardingRegistry.fallback_env** option on the Configuration server.
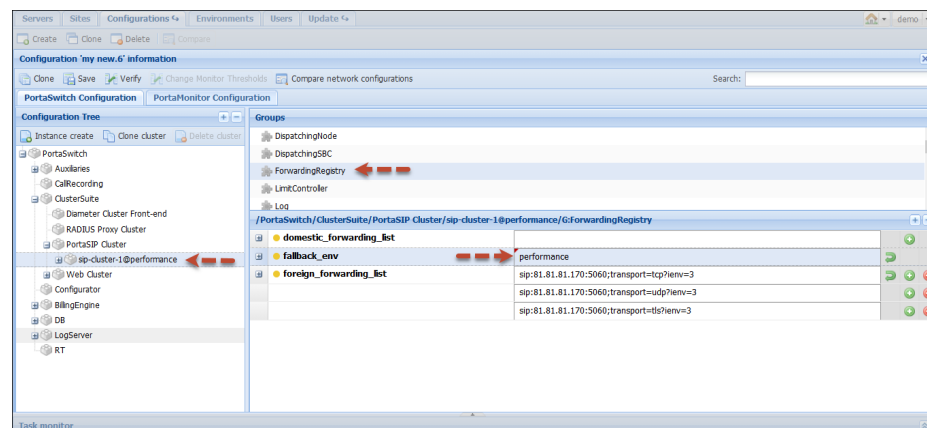
Then when the dispatching SBC cannot identify where to dispatch an incoming request, it routes this request to the source system for further processing.

Fallback call delivery increases the call success rate in Dual Version PortaSwitch® and therefore improves customer experience.

Proxying Diameter requests

In mobile networks, PortaBilling® operates as the OCS (Online Charging System) and communicates with the mobile core 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 target system serves as the inter-system proxy. It accepts all Diameter requests from the source system to the target in Dual Version PortaBilling® and detects where in the system the account is currently located. If the account belongs to the target system, it processes the Diameter request itself. If the account belongs to the source system, the Diameter cluster proxies the request to the source system's Diameter cluster and mediates further communication with that Diameter cluster and the SAE-GW.

This solution facilitates customer transfer from the source system to the target and ensures proper functioning of the Dual Version PortaBilling®.

RADIUS Proxy

The RADIUS proxy is deployed on the target system. It receives RADIUS requests for accounts to be charged for a session. It also detects to which system (source or target) accounts belong. The RADIUS proxy then sends the requests to that specific system.

When a RADIUS server processes the request, the response is sent back to the RADIUS proxy. The RADIUS proxy relays the response to the NAS.

This allows external RADIUS clients (e.g. NAS) to function seamlessly with Dual Version PortaBilling®. It also allows the CDR mediator process to run on either system so it can perform file extraction and parsing yet put the charging data records into the system where the account currently resides.

To add fault tolerance to the RADIUS proxy, it can be clustered with the virtual IP address serving as the communication point for the NAS. If, for some reason, the active RADIUS proxy node becomes unavailable, the virtual IP address switches to another node. That node then becomes active and handles RADIUS request processing.

Number porting requests processing

Number porting is widely used within the wireless communications market as it allows users to keep their numbers when switching service providers.

In Dual Version PortaSwitch®, number porting requests are distributed based on the current location of the customer record. Thus, all incoming requests from Neustar arrive to WebDispatcher, which passes them either to the target system or the source one. This ensures uninterrupted service provisioning and facilitates smooth customer migration.

Let's consider how request distribution works for port-in and port-out requests.

Port-out requests

Port-out requests are created automatically, without an administrator's involvement. So when a user wishes to port their number from PortaBilling® to another provider, WebDispatcher receives the port-out request from Neustar. WebDispatcher finds which system the customer's account record belongs to. If the customer has been moved to the target system, it passes the number porting request there. The target system terminates the customer's account record and updates Neustar about its number availability. Neustar confirms the request completion to WebDispatcher, which passes it to the target system.

If the customer's account is within the source system, the whole porting out procedure runs from there.

Port-in requests

When porting in a user's number, the administrator defines the system where this number will be provisioned. Then the respective system interacts with Neustar to complete the request.

Consider the following example:
John Doe, a former customer of EasyCall Ltd., signs up for HappyTraffic services and wants to keep his number 18185557852. The administrator creates the account 120522254200 for John in the target system and initiates a port-in request to Neustar in the target system's PortaBilling®. After Neustar confirms the request, 120522254200 is replaced with 18185557852 and John starts using the services of HappyTraffic.

The distribution of number porting requests across systems within Dual Version PortaSwitch® facilitates gradual customer migration and ensures the uninterrupted operation of number porting functionality for users, regardless of their location.

Online web signup

Online web signup functionality supplied with PortaBilling® enables you to automate the subscription process for new customers to the service. That is, customers can do everything online without requiring any assistance from your administrator.

For backward compatibility, in Dual Version PortaSwitch® newly subscribed customers are created in the source system by default. If you wish to provide online web signup services in the target system and create new customers there, do the following:

- transfer the signup reseller's data and that of all their customers to the target system via Porter, and
- adjust the signup configuration on the source system by defining the IP address of the target system's SOAP server.
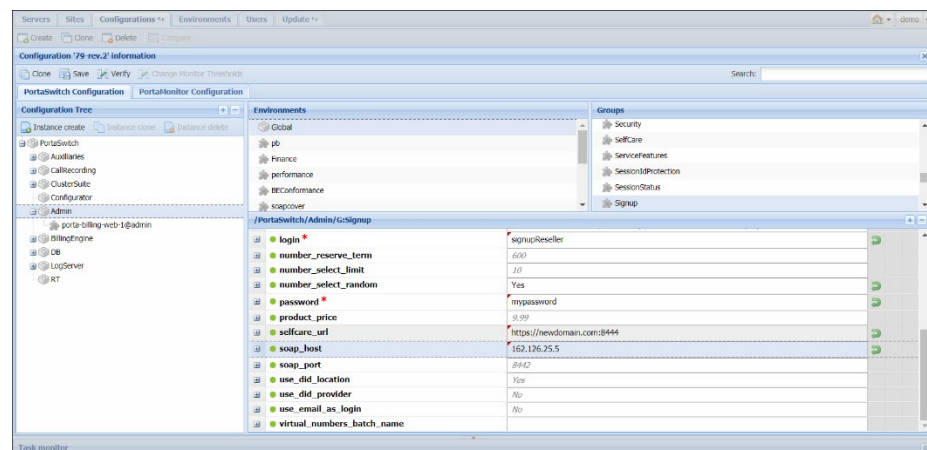
Common API entry point for Dual Version PortaSwitch®

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** to check his balance and invoices, both when his record is on the source system and after it is moved to the target. Administrators and resellers, in turn, must be able to operate in both systems, via the API, without reconfiguring their applications.

Please note that administrators/resellers can log into their customer/account self-care interface. If that customer/account is already moved to the target system, PortaBilling® denies automatic login. This is to protect against fraud, since this login request comes from outside the network. To access the customer/account self-care interface, administrators/resellers must click **Login** again.

To this end, WebDispatcher for Dual Version PortaSwitch® serves as a single API entry point for both the source and the target systems. It accepts API requests from applications (e.g. CRM) and dispatches them across systems for processing.

WebDispatcher is one of the components of the DSBC and operates on the same servers where the DSBC is deployed. Like other DSBC components, WebDispatcher is clustered, runs on several servers and is accessible via a single public IP address.

That is how WebDispatcher works. The application sends an API request to get customer information. WebDispatcher finds which system a customer is located in and sends a request there. After WebDispatcher receives the results, it returns that customer information to the application.
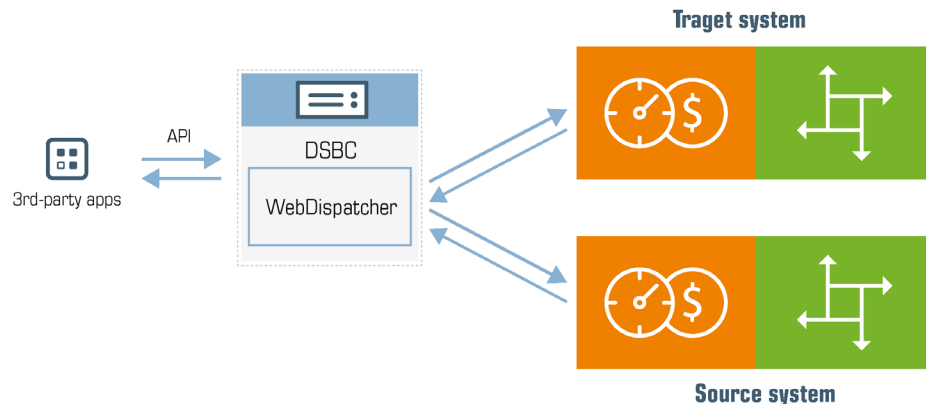
Thus, an application can receive data from the source, the target or even both systems. The decision about which data to retrieve is based on the following:

- Who uses the application – the administrator/reseller or a retail customer/account, and
- The user's location – if it is within the source or the target 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 external self-care portal, the application sends the API request to WebDispatcher. It checks the customer's location within Dual Version PortaSwitch® and if the customer was moved to the target system, WebDispatcher sends the API request to that system.

Then, when the customer performs an action (e.g. selects xDRs for the previous billing period), WebDispatcher proxies the API request to the respective system (e.g. the target system). The target system then retrieves the xDRs and WebDispatcher delivers them to the application.

If a customer in the source system uses the external self-care portal, WebDispatcher receives the API requests and sends them to the source system for processing.

API for administrators and resellers

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

WebDispatcher automatically detects the context based on the i_customer/i_account values passed in the input parameters within API requests. For this purpose, customer and account records in Dual Version PortaSwitch® have unique IDs. When you move customer data from one system to another, IDs are preserved. Later on, customers created on the source system receive odd ID values while customers created on the target system receive even ID values. This prevents data collision during customer migration from one system to another.

Some API applications use both the IDs of customers/accounts and the IDs of entities such as products, volume discount plans, etc. as static values in API requests. For example, to obtain information about a customer's volume discount plan usage, the application sends `i_customer` and `i_vd_plan` values in the input parameters.

To preserve the workflow for these applications in Dual Version PortaSwitch®, IDs for all configuration entities are preserved when moved between systems. Thus, your ABC product has the same ID (e.g. i_product=123) on both systems. As a result, fewer customizations are required to make the API application compatible with Dual Version PortaSwitch®.

Differently from customers and accounts, these entities remain available in the source system, even when moved to the target one. Therefore, the administrator must remember to modify them in both systems in order to prevent differences in configuration.

Once an administrator has configured the interconnection between systems, the workflow is typically the following:

- The application connects to WebDispatcher via the API and receives the session ID.
- If the application sends the API request to retrieve the customer list, WebDispatcher runs the request in both systems and then merges the results into a single list. Thus, the administrator or reseller sees those customers who are still within the source system, those who were moved to the target system and those who were created in the target system.
- To manage a customer in Dual Version PortaSwitch®, the application sends the ID of this customer's record in an API request. WebDispatcher sends the request to the system, where this customer is provisioned. After the 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 target system, the application sets the session context by providing the unique ID for the target system's billing environment within the API request. Then WebDispatcher runs further requests in the target system.
- To operate with subscriptions from the source system, the application switches the session context by providing the ID of the billing environment in the source system. In this case, all subsequent requests are processed by the source system.
- To modify a customer class with the same ID in both systems in Dual Version PortaSwitch® the application must first set the session context to the source system and update the customer class there. Then the application switches the session context to the target system and updates the customer class there. We recommend this approach to avoid creating differences in entity configuration.

CPE profile provisioning is supported for Dual Version PortaSwitch®. So when an IP phone connects to the Internet and requests a configuration file from WebDispatcher, WebDispatcher processes the request and retrieves the file from the target system and then delivers it to the IP phone.

The obsolete TFTP protocol is no longer supported, therefore the HTTP protocol is supported for CPE profile provisioning in Dual Version PortaSwitch®.

WebDispatcher for Dual Version PortaSwitch® provides a single place for customer management and system operation. This makes the migration process more fluid. The full application preserves user habits and thus improves the overall PortaSwitch® customer experience.

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 target 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 WebDispatcher.
5. The common API entry 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 source and the target system.

Known limitations

Bear in mind the following known limitations:
1. For security measures, add the WebDispatcher IP address to the allowed IP addresses for each entity (user, customer, reseller) on the PortaBilling web interface.
2. The systems are mapped with each other by their environment IDs. Therefore, be careful when switching the environment since this may result in broken mapping and session disconnect with the target system.
3. If the target system fails to establish the API session with the source system, it operates as if there is no source system.

Limitations in Dual Version PortaSwitch®

The set of dispatchers in Dual Version PortaSwitch® distribute calls, registrations, API, number porting, CPE, RADIUS and Diameter requests

thus ensuring customers in either system keep using the services as if nothing has happened.

However, some services have limitations in scope of Dual Version PortaSwitch and require additional actions from the administrator. These services are:

- **Mail services.** If a user works with their mailbox via an external mail client, this client will not work properly after the user is migrated to the target system as it will keep sending the IMAP/SMTP requests to the source system's address. The Mail proxy component operates only with local requests and does not support IMAP/SMTP request distribution across systems. Thus, it is required to reconfigure the mail clients to send IMAP/SMTP requests to the URL of the target system.

- **SMS delivery via SMPP.** Like mail proxy, the SMPP proxy operates locally, i.e. it processes SMPP requests in the system it is configured in. The SMPP proxy does not support the SMPP request dispatching across systems in Dual Version PortaSwitch. Therefore, agree with your SMS providers to establish TCP connections with both source and target systems to handle SMPP traffic from them.

- **Access to webmail and PortaSIP API.** Migrated customers can access their web mailboxes from their account self-care portal by clicking the Voicemail Inbox button. The direct access (i.e. by entering the web server's URL in the browser) is not available until customers are provided with the IP address/domain name of the target system. Similarly, to access PortaSIP API (e.g. to configure the auto-attendant), their API applications must be adjusted to send requests to the IP address/domain name of the target system. This is because these API requests are processed by a separate SOAP server which does not support request distribution.

- **Callback services.** Web and email callback services require reconfiguration to become available for migrated customers. Namely, provide customers with the email address and web page address to initiate email/web callback in the target system.

- **Call control API.** For customers to receive to real-time call state notifications, a WebSocket connection must be established and the API applications must subscribe to notifications via this connection. WebSocket API is not supported in Dual Version PortaSwitch®. Thus, though your customers can still use XML

(SOAP) and REST (JSON) API for call control API methods, they will not receive call state notifications.

- **IVR applications**. IVR applications with access numbers defined as DIDs are available only if all customers using them reside in the same system. Thus, to preserve the service flow you must migrate all of them and the access number at once.

Frequently asked questions

What is the difference between Zero-downtime update (ZDU) and Dual Version PortaSwitch®?

ZDU allows to update the whole system to the new software version in a single step. Dual Version PortaSwitch® solution implies the following:

- Simultaneous operation of two independent systems in different software versions linked with each other.
- Consistent migration of customers from a system with a lower software version to a higher version system.

Using the Dual Version PortaSwitch® solution, you can gradually migrate customers from older releases to a newer release in 2-3 months (exact time depends on the database size).

The maximal update "jump" span for ZDU is 5 releases because the nearest LTS cannot be crossed in such a "jump." Maximal "jumps" using Dual Version PortaSwitch® solution are much wider but require obligatory checkpoints every 15 releases (i.e., MR55→MR70 or MR70→MR85).

Does the Dual Version architecture provide redundancy?

The target and source systems of the Dual Version solution operate independently and do not involve redundancy as part of the design. If a data center outage occurs for either the source or target system during a long migration, it's going to be visible. So the Dual Version solution does not provide redundancy by itself, but instead works in conjunction with the current geo-redundancy and ZDU functionality available in PortaSwitch®.

Since Dual Version PortaSwitch® involves two systems running in parallel, the source and target systems can be used for geo-redundancy to ensure uninterrupted service provisioning for every system in Dual Version PortaSwitch®.

In most cases, the redundancy can be achieved in two ways: by deploying billing, SIP, and database clusters, or by dispersing the systems across multiple sites. The latter solution provides high-availability and geo-redundancy options.

Do I need to deploy a dispatching SBC (DSBC) if I already have the PortaSIP® cluster?

Yes, you always need to have a separate DSBC for Dual Version PortaSwitch®. Even if you have a fairly new PortaSIP®, it's still a part of your source system and is only aware of that specific system. If you want to move, for instance, from MR70 to MR85, you need a DSBC in front of your PortaSIP®. DSBC decides whether to route calls to the PortaSIP® cluster on the source system or the target system.

The DSBC links systems in Dual Version PortaSwitch® and is required for delivering calls across systems. It "knows" where the account record is located and delivers the call accordingly. You can configure DSBC on a real or virtual server or in the PortaOne Cloud.

Does the DSBC participate in media streaming?

No. The DSBC only participates in the signaling path of the call by distributing incoming call initiation requests across systems for processing. When a call is established, endpoints exchange media streams directly (except for calls with RTPProxy, where the media stream goes through SIP Cluster without involving the DSBC), or it can take place via the RTPProxies of their respective PortaSIP® clusters.

Where can I deploy the target system?

You have the choice to either:

- Use dedicated hardware (on-premise or virtual).
- Launch the target system fully in the PortaOne Cloud.
- Use the PortaOne Cloud for the DSBC and the dedicated hardware (on-premise or virtual) for the rest of the target system.

The cloud can be a temporary migration tool, or you can use this as an opportunity to liberate your operations from the physical data center permanently. At PortaOne, we consider the cloud to be the way to move forward and future-proof your operations. However, to limit latency, we recommend deploying the DSBC (by default, deployed on the target system) as geographically close to the source system as possible.

82

Where do I need to install the dispatcher node: target or source system?

All dispatcher nodes need to be deployed within the context of the target system installation and configured by the target system's Configurator.

What are the licensing terms for using Dual Version PortaSwitch®?

An extra system requires extra licensing. The licensing depends on how you wish to deploy the target system. You can purchase a regular perpetual license and use it for your target system in the Dual Version PortaSwitch® solution or as a separate system to ensure redundancy – that's up to you. In the case of PortaOne cloud deployment, you benefit from the monthly fee calculated according to the number of your billable xDRs processed per month.

If you're only interested in extending the system for a one-time migration, we can come up with an exclusive license to cover this situation. For detailed information about pricing, contact the PortaOne Sales team.

Can I launch Dual Version PortaSwitch® if my current release is lower than MR50?

No. The Dual Version PortaSwitch® solution is only available starting from MR55-6. If your current release is lower than MR55-6, you need to update to MR55-6 first. Contact PortaOne Support to schedule the update procedure.

Which releases are available for "long jump" via Dual Version PortaSwitch®?

Currently, the "long jump" options are available for MR55-6 to MR70-6 and MR70-6 to MR85-3.
For detailed information about the transfer capabilities available for your current release and build, as well as for assistance in performing the transfer, contact PortaOne Support.

Can I run a pre-test to ensure that my customers are migrated successfully before I start the migration process?

Yes, to a certain degree, by using the "dry-run" mode of the Porter utility. In this mode, Porter emulates (there will be no new records in the database on the target system) exporting all data related to the customer and applies transformation logic for this data. Therefore, with the help of Porter, you can check the configuration of the target system before running the migration process.

Is there any downtime when a customer is moved between systems?

Yes. A customer can experience downtime ranging from several seconds to several minutes (for a customer with more than 100 accounts) while being migrated.

The migration of *customer data* related to the current configuration creates some downtime, but the migration of customer's history of service usage (xDRs, volume discount counter history, call recordings, etc.) causes no downtime. During the customer data migration, services (LTE, SIP, xDR mediation, etc.) are affected by the downtime as the customer data might already be deleted from the source system, but not yet created on the target system.

Customer migration time strictly depends on the amount of data being migrated – for instance, the number of accounts a customer owns: it takes less time to migrate a residential customer with a basic SIP calling service than an IP Centrex customer with dozens of accounts.

How many customers can I batch migrate at a time?

You can batch migrate any number of customers at a time. The number of customers in a batch is arbitrary. And the time it takes to migrate the batch depends on the number of accounts per customer and the overall capacity of the given Dual Version PortaSwitch® installation. For example (based on real customer data, approximated):]

Customers	Accounts	VDCH discount history records	NAUD Internet session records	CDR_Accounts/CDR_Accounts Failed	Migration time (seconds)
1	5	15	330	255	8
1	19	236	0	1375	105

Several parallel threads deal with entities in the batch to provide maximally fast customer migration thanks to the new smart Porter utility. Each thread processes its customer and starts transferring the data belonging to that customer. In this process, threads also branch into parallel "workers" that work in a single customer's context, reducing the overall migration downtime.

For example, if the customer has 4 accounts and 4 workers, then each can take one account and start transferring its data to the target system. Threads allow you to port several customers at once, thus making better use of the hardware. These parameters are selected automatically (and not by an administrator). You can try and pre-define them, but the automatic way is recommended.

Can I migrate just a part of the reseller customers?

Yes. Partial migration allows you to migrate just a small batch, see how it behaves, test, and do a rollback in case of errors or UX problems. In this case, the reseller is not marked as Exported and thus exists in both systems. This implies reseller management in both systems.

Can I add new customers, products, etc. to the systems during data migration?

On the source system, you can add new customers during data migration, but you cannot add new products. On the target system, you can operate independently from the source system and create new products and service bundles, evaluate new features, etc.

Will vendors be marked Exported after migration to the target system?

No. The vendor configuration is duplicated in Dual Version PortaSwitch®. This means that the same vendor is active in both systems and their balance is split between the systems.

What happens to the vendor's balance on the source and target systems?

The same vendor is active in both systems, so the balance is split between the source and target systems. The vendor's balance is affected by the calls and services terminated on this vendor on each system. So, while the customers use the services on different systems, the vendor's balance on the target and source system will not match.

For example, if the vendor's balance on the source system is $2000, initially it will also be displayed as $2000 for the target system. However, as some customers are migrated to the target system and some stay on the

source system, the balance for the services used will be displayed differently on different systems: $2000 + the price of the services used on the source system and $2000 + the price of the services used on the target system. This means that until all customers are fully migrated to the target system, the vendor's balance on source and target systems will be different.

Can I modify a product that is only used by accounts on the source system in Dual Version PortaSwitch®?

No. Product data is dictionary data, which must be in sync on both the source and target system at all times.

If a product is modified on the source system, the service behavior of an account will change after migration to the target system, leading to undesirable results because the system doesn't expect such behavior. If a new product is created on the source system and then assigned to some account, the migration of the customer with such an account will fail.

Everything that can be changed even before customer creation is dictionary data and should always be synchronized on both source and target systems. Remember, you can create or change Products even before creating a customer.

Can I change the product configuration of accounts on the source system in Dual Version PortaSwitch®?

Yes. The set of products assigned to accounts is customer data and it and can be safely changed at any time. Any configuration of products created before switching to Dual Version PortaSwitch® can be used.

How many public IPs do I need to deploy a DSBC for Dual Version PortaSwitch®?

The minimum requirements are:

- 1 IP address for Web Dispatcher*
- 1 IP address for RADIUS/Diameter Dispatcher
- 1 IP address for each SIP Cluster on the source system

* Web Dispatcher and the RADIUS/Diameter Dispatcher could technically share the same IP in the context of DSBC, but that would bring inconveniences to customers. For example, if they currently direct the RADIUS/Diameter traffic to **IP address 1** and the Web traffic to **IP address 2**, you can deploy the DSBC where the Web Dispatcher and the RADIUS/Diameter Dispatcher nodes are both using **IP address 3**. In

this case, all the customers need to reconfigure their settings and re-route both their RADIUS/Diameter and Web traffic to **IP address 3**.

Or you can dispatch the DSBC using **IP address 1** for the RADIUS/Diameter traffic and **IP address 2** for the Web traffic (consequently, you will need to allocate new IP addresses for these nodes on the source systems and later on the target systems) and the customers won't even notice that anything has changed in the system, as the DSCB will route the traffic incoming to **IP address 1** and **IP address 2** to the corresponding new IP addresses on the source and on the target systems.

How much time does it take to prepare everything for the migration (time to deploy, configure, prepare the source and target systems for the migration)?

This depends on many factors. The isolated technical procedure of deployment consists of running just a few commands because in the new Dual Version PortaSwitch®, everything is maximally automated. In the testing lab, the full preparation process only takes a few days. However, in a real-world setting, the whole preliminary testing stage combined with the business analysis stage might take a long time (up to several weeks).

What about the performance figures: how much time does it take to migrate my database?

Migration with the help of the Dual Version solution can be a very swift process, but we're striving to strike the balance between your comfort and speed. You can take as long as you need for a gradual controlled migration/update. However, the minimum time it can take to migrate everything with all xDRs, even for the largest client data sets, is around 2 months.

I have custom reports, templates, music on hold files, and a few other data types. Will it be automatically migrated, or will it take some manual work?

It depends on the data type. The Porter utility moves all customer data (subordinate customer configuration data that belongs to a specific customer and can only be created after that customer is created, for example, accounts, xDRs, voicemails), but doesn't migrate the dictionary data (the data that initially exists on the source system, independent from customer creation – for example, products, templates, customer class) as it's assumed that it won't be changing.

How do I migrate voicemails in Dual Version PortaSwitch®?

Voicemails are a part of customer data. They are automatically migrated by the Porter utility during the customer migration process.

How do IVR numbers work in Dual Version PortaSwitch®?

Dispatching of the calls to IVR numbers during migration with the help of Dual Version PortaSwitch® is supported for the IVR numbers that have corresponding DID numbers assigned to them. The DSBC routes the incoming calls to such IVRs to the appropriate system (source or target).

However, in some cases, IVR numbers are used without a DID number assigned to them. For such IVR numbers, during the migration, the DSBC won't "know," to which system to address the calls. To prevent this, such IVR numbers need to be migrated together with the product, customers, and accounts that use these IVR numbers.

What happens with the xDR import process during the customer's migration in Dual Version PortaSwitch®?

The same logic as for other services that take place during the migration process of the customer applies here. Customer migration time is a downtime period for services. The import of xRDs for the customer in migration will fail, but the failed xDRs can be selected and set for reprocessing.