PORTA ONE

Porta Switch®

Architecture and Concepts

Maintenance Release 40

www.portaone.com

Copyright Notice & Disclaimers

PortaSwitch Architecture and Concepts, July 2014
Maintenance Release 40
V1.40.2

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

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`

The **exclamation mark** draws your attention to important information or actions.

NOTE: Notes contain helpful suggestions about or references to materials not contained in this manual.

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

Tips provide information that might help you solve a problem.

Trademarks and Copyrights

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

Hardware and Software Requirements

Server System Recommendations

Seven (7) UNIX Servers for the PortaSwitch® or thirteen (13) servers for PortaSwitch® Procinctus. For additional details regarding the recommended hardware configuration of each server consult the **Hardware Recommendations** section on our website.

For information about whether particular hardware is supported by Oracle Enterprise Linux from the JumpStart Installation DVD, consult the related document on the Oracle or RedHat website:
https://hardware.redhat.com/

Client System Recommendations

- OS: Windows XP, Vista or 7, UNIX or Mac OS X
- Web browser: Internet Explorer 7.0 (or higher), Mozilla Firefox 3.6 (or higher)
- JavaScript and cookies enabled in web browser
- Spreadsheet processor (MS Excel or OpenOffice Calc)
- Display settings:
  - Minimum screen resolution: 1024 x 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 100KB and billing logs by approximately 50KB.
- Each registration processed by PortaSwitch® will increase SIP logs by approximately 10 KB and billing logs by approximately 5KB.

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® Switching 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 10MMM (ten million minutes per month). Taking into consideration that for every failed call, logs and CDRs 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
- *The PortaSIP® Switching Server:*
  - 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 CDR takes up about 1.5KB database space.
- PortaBilling® produces at least 2 CDRs for each call. Actually, the number of CDRs produced depends on the call scenario and can reach up to 5-7 or even more CDRs for complex calls. For simplicity's sake, we assume that on average, 3 CDRs 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 CDRs for the previous 60 days.

Using the figures and assumptions described above we can estimate the disk space that will be consumed to process 10MMM (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 10MMM.

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 Switching 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, 1GB of disk space is required).

Installation

A jumpstart installation DVD is provided for all PortaOne products. This DVD contains installation media for Oracle Enterprise Linux (64-bit version), supplementary packages necessary for convenient system administration and maintenance, and all required software packages. After the installation is complete you will assign roles (e.g. RADIUS, web interface, etc.) to individual servers using the configuration server tool – this will automatically enable the required components of PortaBilling® 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 providing communication services such as hosted IP PBX, prepaid card calling, broadband/wireless Internet access and others. The main components of PortaSwitch® are:

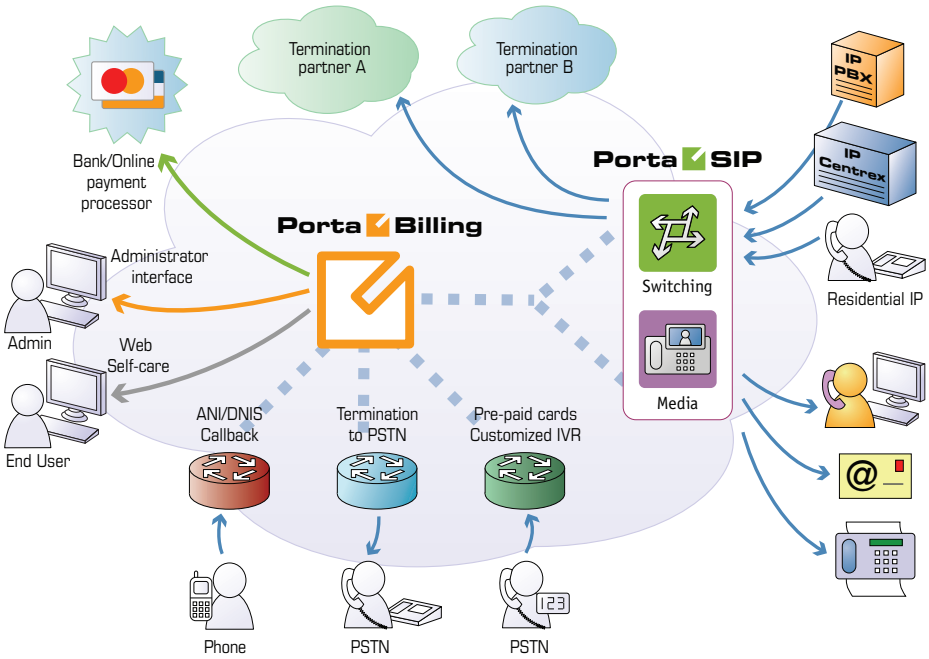
- PortaBilling® – real-time converged billing and service provisioning system
- PortaSIP switching node – class 4 and 5 softswitch
- PortaSIP media server node – media server for IVR applications

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

The PortaBilling component stores all the information about your products, rates and customers and the service configuration for individual customers or phone lines. It is managed via a web interface and includes a self-care portal for your end-users. PortaBilling® controls call processing on PortaSIP (whether a caller is allowed to make a call) as well as real-time routing (which carriers and in which order should be used to send a call in order to maximize profit and provide the required quality of service). PortaBilling® is a converged system; it can be used as a single administration interface to manage (or bill for) multiple services, including those provided by third-party network elements (for instance, LTE SAE-GW or WiMAX ASN-GW), while charges for the different services will be grouped on a single bill.

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

As a service provider, you want not only to let customers make phone calls, but also to use the platform as your main source of revenue generation. Thus PortaSwitch® provides real-time verification of available funds, detects and prevents potential fraudulent activity, automatically disconnects calls to prevent balance overdrafts, provides flexible rating of service usage, offers a tool to create attractive product bundles, automatically assesses monthly recurring charges, generates and delivers invoices electronically, and enforces the payment collection process.



Centralized Configuration Management

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

Configuration server carries out exactly this task, providing an interface for the administrator to view the current configuration, create a new configuration and correctly apply it to all servers, or rollback to an old configuration if a problem has been detected. Another important role of the configuration server is that it stores "images" of different versions of the software. Each image is the actual content (in a binary format) of a specific version of the software code (e.g. Maintenance Release 39, 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 (two servers) or a full PortaSwitch® Procinctus (twelve servers). In the rest of this section, we will use some examples related to

managing the full PortaSwitch® configuration, 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 an application component (e.g. RADIUS server) configured in a particular way and running on a server, i.e. it is a combination of the software code, configuration data, and running processes that provides an actual service. For example, a PortaBilling® RADIUS instance with IP address 1.2.3.4 may be created on server ABC in order to process authentication requests. Or three instances of PortaSIP® 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.

Option

An option is a configuration parameter which alters the system's functionality. Some examples of options would be "Which IP address should the PortaSIP® service run on," "When should statistics generation be done," or "Should the previous balance be included in the invoice's amount due". The value of an option could be a on/off switch, one of the choices made via a select menu, or a text string.

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.

On the top level of the tree are the global options, i.e. those that have an effect on all components of the system.

Below the global level is the role level. Each role is presented as a separate node in the tree, so in order to change the option "When should statistics generation be done" (which is related to the PortaBilling® invoicing role), you would first open the "Invoicing" node.

The next level is the environment level. This provides control for options specific to a particular virtual environment. Finally, there is the instance level, which covers options related to a specific instance (e.g. start accounting is enabled on the PortaSIP® instance with IP address 1.2.3.4).

When the value for an option is defined on some level, it is automatically propagated to all levels below; e.g. if the "start accounting" option is activated on the global level, start accounting will be active by default for all PortaSIP® instances in all environments. It is possible to override options on lower levels; e.g. in our previous example, "start accounting" can be selectively disabled for a specific virtual environment or a particular instance.

Since normally there are many individual options available on each level, it would be inconvenient to work with them if they were organized into a single long list. Thus options are split into groups, with each group containing a small set of options related to a single software module or feature.

Multiple configuration trees

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:

- Clone the current configuration tree into a new one;
- Make the required changes;
- 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.

Deploying PortaSwitch® Across Multiple Sites

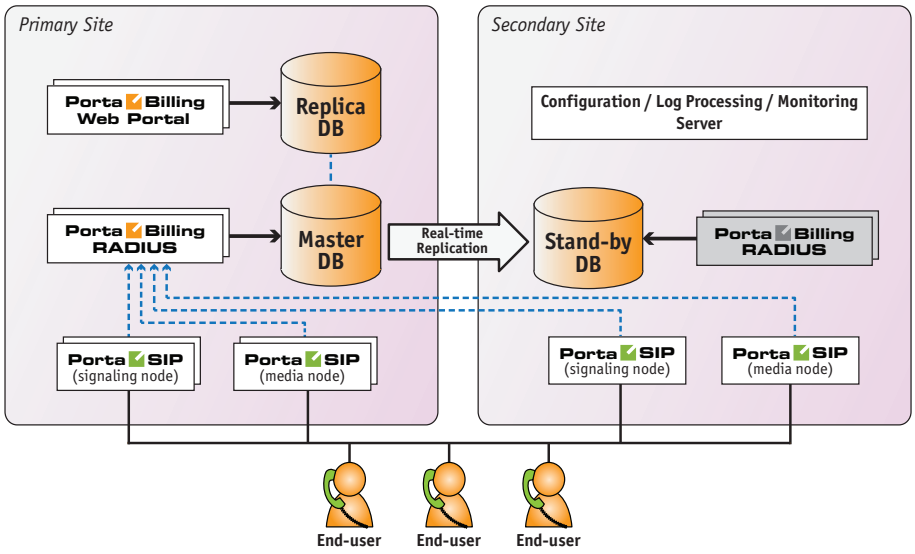
To meet customers' expectations regarding the quality of communication services the service provider needs to introduce an extra degree of reliability within the network and its applications, so that the service is not interrupted – even if some network components are not functioning. How can this demand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

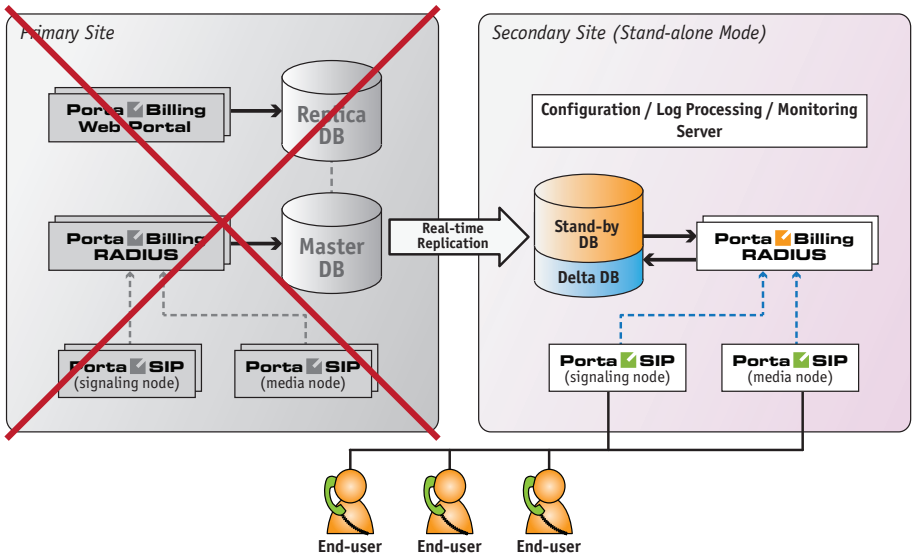
It is important in this situation that the "secondary" site not only activates and begins providing service as soon as possible, but also that it automatically synchronizes the changes later on (updates balances, xDRs, etc.) to the "main" site.

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

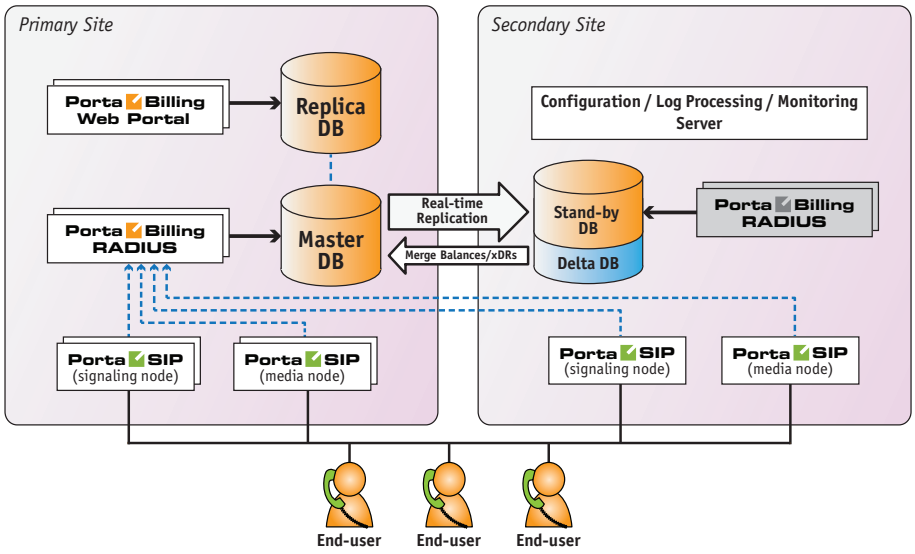
- protect themselves against hosting facility outages
- provide service to multiple geographic regions – even if network connectivity between those regions is lost
- and finally, perform upgrades to new software versions with zero downtime! This last provision adds an essential benefit to the deployment 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 update.



So if the secondary site detects that the main site has become unavailable, the "stand-alone" mode is activated on the secondary site and now it provides the service to end-users using the latest available snapshot of the service configuration. The xDRs for consumed services and changes in balance are accumulated in a separate database (on the stand-by database server) and are taken into consideration when authorizing subsequent activities, so there is no risk of balance overdraft when the stand-alone mode is used.

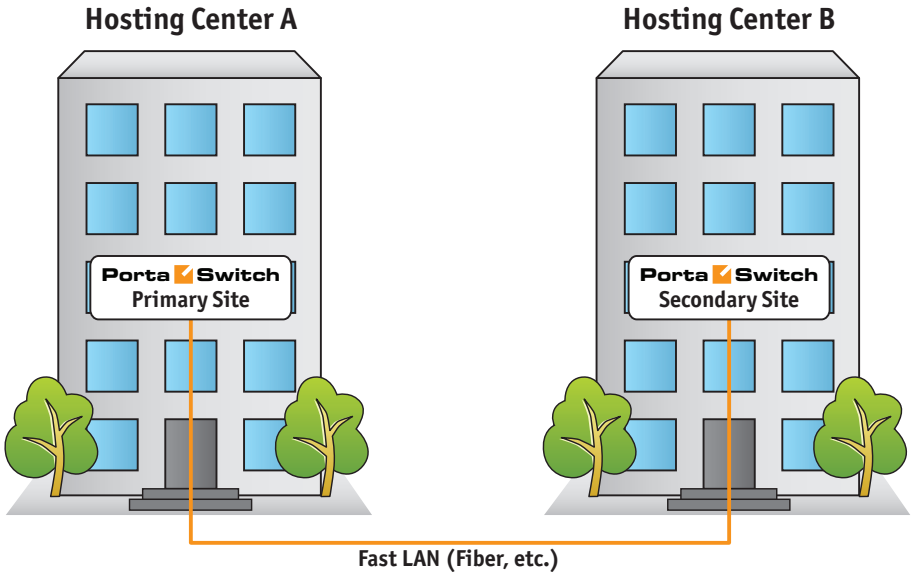


Once the main site becomes available again, the secondary site starts the process of synchronizing all of the accumulated changes to the main site and then the secondary site switches back to its normal ("stand-by") mode.



Typical Deployment Scenario

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

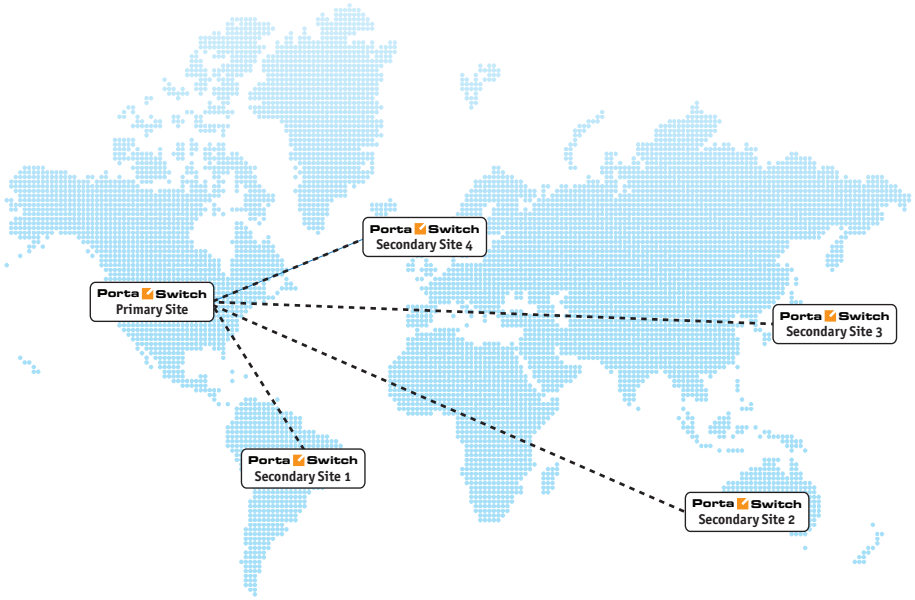


Within its "normal" mode of operation at the remote site:

- The stand-by database server continually retrieves changes from the main site, so it always has an up-to-date snapshot of the database from the main site.

- The RADIUS billing servers are in "stand-by" mode, so they do not actively process any requests.
- The PortaSIP® servers provide service as usual (processing incoming calls, playing the IVR, etc.). They use the RADIUS servers on the **main** site for authentication and write any changes (e.g. updated SIP phone location) into the primary database.

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



When Disaster Strikes

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

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

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

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

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

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

Example Scenario

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

- A power outage makes the entire primary site unavailable.
- This event is detected by a watchdog script on the secondary site so it switches into "stand-alone" mode (in particular, this enables the RADIUS server on the secondary site and instructs PortaSIP® servers on the secondary site to use it as the authorization source).
- If the user's SIP phone was previously registered to the PortaSIP® server on the primary site, during the next re-registration attempt the phone will detect that the server 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® server on the secondary site it registers there. (If the phone is already registered on the PortaSIP® server on the secondary site, nothing changes.)

- When the user attempts to make an outgoing call, an authorization request is sent to the PortaBilling® RADIUS server on the secondary site.
- The billing engine uses the currently available balance information ($98.00) to compare it with the credit limit ($100.00) and authorizes the call for no more than 20 minutes.
- When, after 12 minutes of conversation, the user hangs up, PortaSIP® sends an accounting request to PortaBilling® so that charges are applied.
- When PortaBilling® processes the request, it calculates the amount to be charged ($1.20) and stores the balance adjustment ($1.20) and the xDR for that call (with all call details such as CLI, CLD, call connect time, etc.) in the delta database.
- Then, when the user makes another call and PortaSIP® sends an authorization request, the billing engine calculates the "effective" balance as the sum of the balance in the stand-by database ($98.00) and the balance adjustment stored in the delta database ($1.20). So the effective balance is $99.20 and the call will have a time limit of 8 minutes.
- The user hangs up after 5 minutes, so there is another xDR for that call with the charged amount of $0.50 written to the delta database and the balance adjustment is now $1.70.
- The next call will only be authorized for the remaining $0.30 of available funds – and can only run until the balance reaches the credit limit. This prevents balance overdraft – even if the site operates in stand-alone mode and the balances in the stand-by database are not changed.
- When the primary site comes back up, synchronization takes place.
- First to happen is that funds in the amount of the balance adjustment ($1.70) are locked in the primary database – this ensures that if a customer now tries to use the service on the main site, he will only be able to spend the $0.30 that he has available.
- Next, the secondary site is switched back to "normal" mode.
- And then, individual xDRs are transferred to the primary database.
  This two-step process (first funds lock, then actual xDR transfer) ensures the avoidance of balance overdraft on the main site while an xDR transfer is in progress. There can be a large number of xDRs (if a secondary site operated in stand-alone for an extended period of time) and consequently, it can take time to replicate all of them to the primary site.

Stand-alone Mode Restrictions

First of all it is important to understand that from the point of view of the secondary site it is impossible to differentiate between two types of events:

- Primary site is down or destroyed (power failure, hurricane, earthquake, etc.)
- The primary site is still up and operational, but the 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 cities (sites), each site functions normally – there are users in each city using the service.

Data integrity between primary and secondary sites must be protected at all times, i.e. no operations should be allowed to run on the secondary site that could cause data conflict when merging the changes back to the primary site.

For instance, if during a connectivity outage between the sites:

- The end user (connected to the secondary site) changes his service configuration (e.g. sets up call forwarding to phone number 123)
- In parallel, the administrator also changes the forwarding settings on the primary site (and sets it up to 567)

What should happen during the data merge when connectivity is restored? It would be unclear that what remains is a valid configuration (i.e. which number should end up as the forwarding number). This is called a "split brain" problem and, of course, should be prevented from happening.

So although the secondary site can detect that the primary site is not accessible, it always considers the possibility that, in fact, the primary site is operating normally, there are users making calls, administrators making changes on the web interface and that 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. So while the secondary site is in stand-alone mode, users can make and receive phone calls (using IP phones or calling card IVRs) and use IVR applications that do not change the configuration (e.g. announce the current balance). Web interface and IVR applications such as account self-care are disabled.

Updating the System to a New Version

Updating your system to a new release (whether it is your personal WiFi router or a powerful PortaSwitch® server, serving tens of thousands of customers) is always a challenging task. You ask yourself questions such as: How long will it take? Will updates be applied correctly to all system components involved? What happens if something goes wrong – can I get my system back to the operational state it is in now, etc.?
PortaSwitch® utilizes an innovative system of maintaining and modifying the software code on each server, allowing you to properly address all of the concerns expressed above and ensure that you are able to:

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

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

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

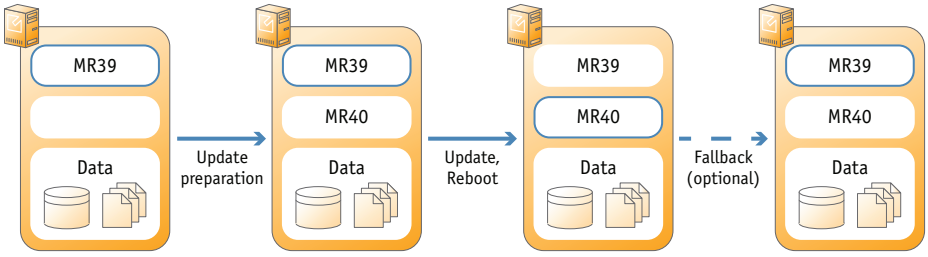
This is why PortaSwitch® utilizes a dual software version management system that has none of the weaknesses described above.

The disk subsystem on each PortaSwitch® server contains three (3) separate partitions. One of them is used to store the actual application data, i.e., database files, logs and .CSV files with statistics of the customer's activity, etc. Two (2) other partitions are equal in size and each of them can contain the full set of the software "code" required to operate the server – operating system, third-party libraries and modules (e.g. Apache) and the actual code for a specific application, e.g. PortaSIP®. At any given moment one of these partitions is considered active: this means that upon startup, the server uses this particular partition to boot up and the application code, located within it, is used to

operate the service. When the system is being prepared for an update to a new release, the other partition is cleared and the new version of the code is installed there. This is done while the system is still operating under the current version of the software, without any service interruption. Now the server has all the required data to operate with the new release – moreover, since the new release is installed as a set of binary packages, one can be sure that this is exactly the same code (the same version of operating system, the same version of kernel and the same bytes in every single utility or file!) that was used in PortaOne's labs during the testing period, that was deployed on staging systems during the field testing and that is currently being used by other PortaOne customers worldwide.

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

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



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

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

Updating the Application Data

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

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

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

So the update process includes two extra steps:

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

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

The PortaOne team has a development and testing process specifically aimed to make this possible, but we can only guarantee this inter-operability for the adjoining releases. In other words, the system operating with MR39 will operate normally while the data is being updated to MR40 (or it is possible to rollback to MR39 after the migration to MR40 has been done). It is not possible to provide this transparency when the distance between releases is too great (e.g. MR36 will most likely *not* operate properly if the data has already been updated to the MR39 format). Thus, the preferred method of updating that provides unlimited time for update preparation and allows for fallback to the previously used version is to go step by step: e.g. from MR39 to MR40, then from MR40 to MR41 and finally from MR41 to MR42, etc. (If required, it is still

possible to do a MR36 to MR39 update in one go, of course – but in this case there is no possibility of performing a rollback).

Zero-downtime Update

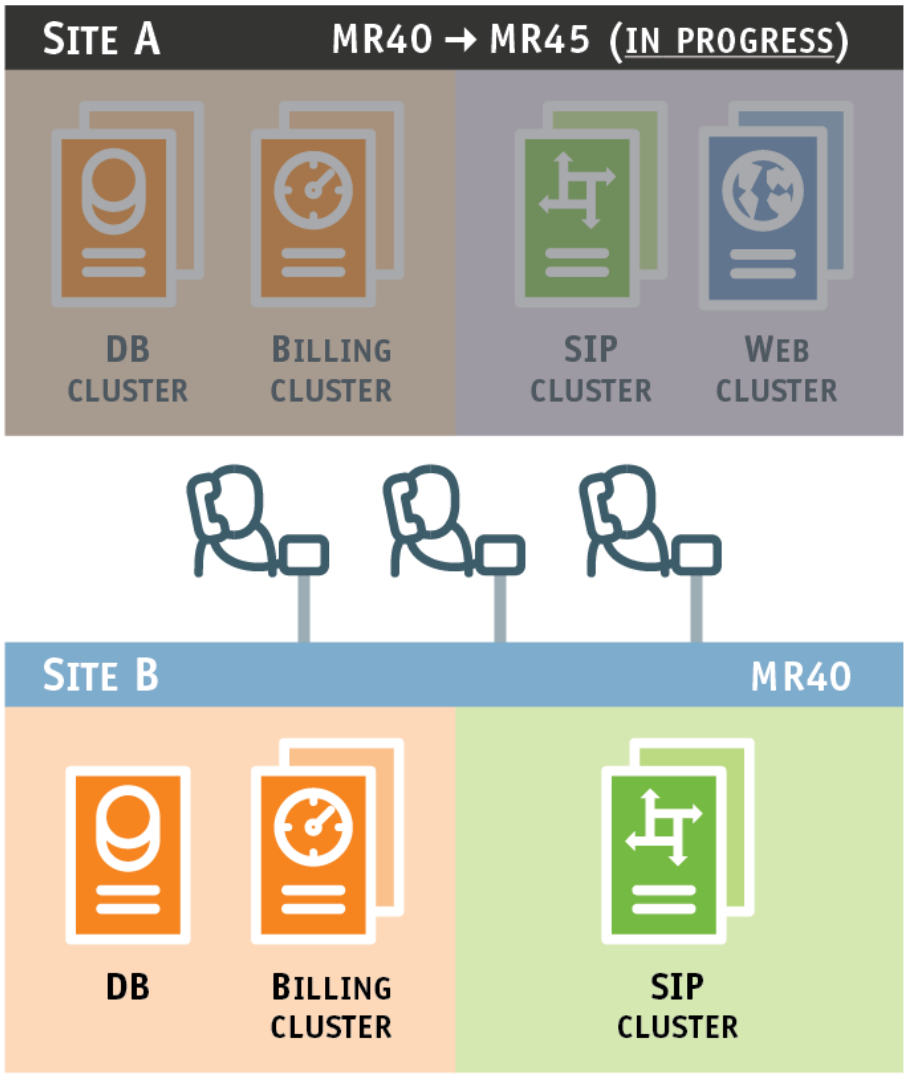
As a result of PortaOne's agile development cycle, new maintenance releases are delivered every 2 months, and each new release contains numerous enhancements that enable new or improved services. Naturally telco operators want to maintain an up-to-date system to uphold their competitive edge – and this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

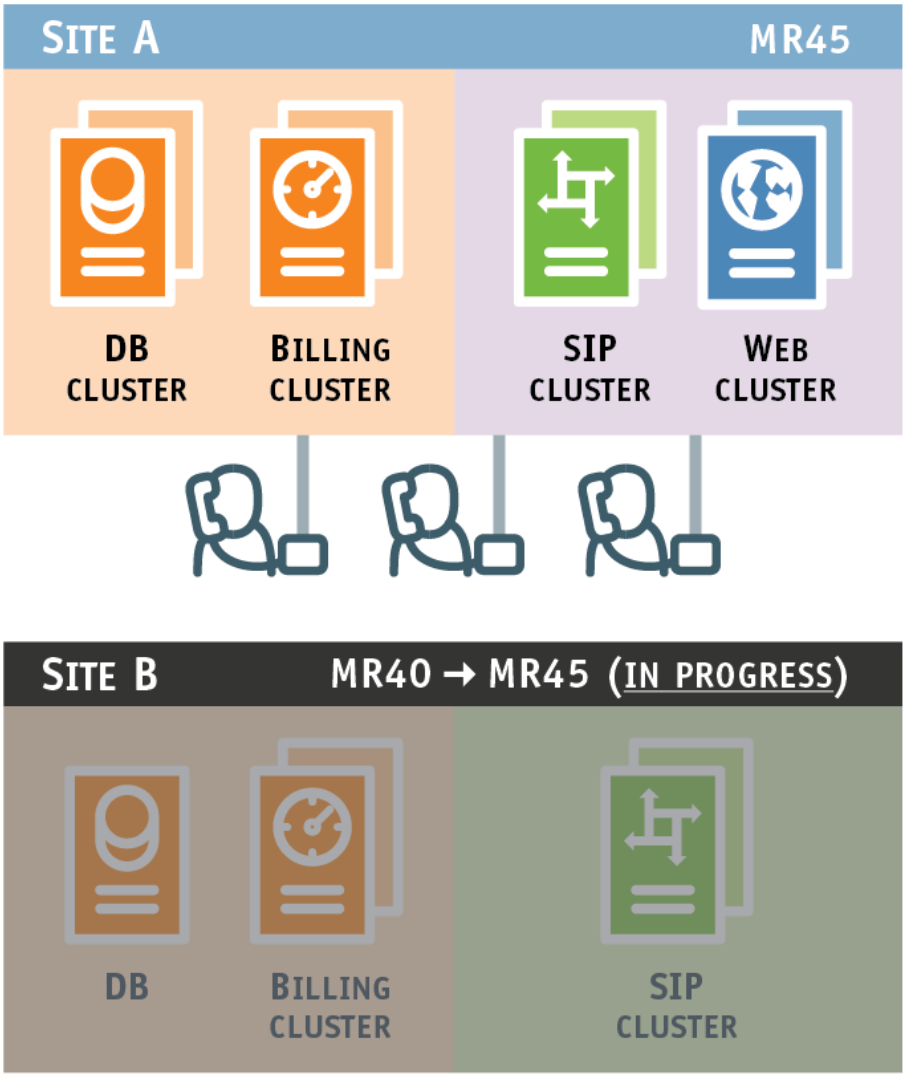
Zero-downtime update (ZDU) technology allows you to perform software upgrades at any time without any noticeable impact on end users.

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

- First, the usual upgrade preparation is done (verification of custom modifications, hardware check for compatibility with new OS, etc.).
- Switching customers to site B will be done via modifying DNS records, redirecting from an external SBC and altering IP routing or other applicable technology. Prior to the start of the actual upgrade procedure, it is highly advisable to test the switch-over to PortaSIP servers on site B (while both sites are running). This is an optional step just to verify that the switch tools work as anticipated outside of PortaSwitch.
- Then at the scheduled time, when secondary site B is switched to the stand-alone mode, the upgrade begins. All new client traffic is relocated to site B (as described above).
- After all existing calls or sessions are completed on site A and there is no live traffic there, the upgrade process begins on that site. Database changes are applied, a new version of code is installed and servers are rebooted.
- Throughout this time, customers are able to use services via site B as usual; calls / messages / sessions are charged, balances are updated accordingly and xDRs are being written. The only restriction during this time is that administrators or customers cannot change their service configuration via the web, etc.
- Finally, when the upgrade is completed on site A and all services there are verified to be working properly, site A is activated. Then the synchronization process with site B begins, and changes in balances and xDRs accumulated on site B while in stand-alone mode are applied to the main database on site A.
- All new customer calls / sessions are directed to site A (using the same tools as described earlier). Any calls in progress on site B will complete as normal.

- When site B has fully synchronized its data with site A and there are no more "live" calls there, the update process begins on site B. It replicates all changes in the database structure from site A, a new version of the software is installed and servers are rebooted.
- Finally, site B returns to its standard operational mode: PortaSIP servers there may be used in conjunction with the main site to process calls; the billing servers and database are in stand-by mode and data changes on site A are immediately replicated to site B so it has an up-to-date copy of all service configurations.





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

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

Custom Modification Management

To compete with larger incumbent telco operators, independent service providers must not only provide the functionalities that end-users want faster than anybody else on the market – but also provide these functionalities exactly how the end-users anticipate seeing it. The first condition is perfectly implemented by PortaOne and its agile development

process, which allows PortaOne to offer an incomparably short waiting time for a new functionality. The second condition can be achieved by adjusting standard functionality according to the unique needs of your customers.

Modifications to standard functionality can be both light and solid. But in both cases you will need to introduce changes to the software installed on your servers and maintain them through software upgrades. Usually, the more modifications you introduce, the more difficult they become to maintain. To simplify this process, PortaSwitch® provides convenient tools for managing your custom software, patches and files:

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

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

Per-configuration Licensing: Flexibility and Control

A license verification method is based on centrally distributed **license files** and provides the user with a more flexible and convenient service management system. From now on, applications (e.g., PortaBilling® RADIUS or PortaSIP®) are no longer bound to a specific physical server by a dongle. This enables you to change the system configuration and add launch applications on new servers without manually reconnecting the dongles or, in fact, without any need for your physical presence in the facility where your servers are located.

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

The time required to deploy new applications after the license has been purchased has been dramatically reduced since you no longer need to wait for the delivery of a dongle to put your acquired licenses to work. For example, if a rapid increase in traffic is anticipated this coming weekend, you can contact the PortaOne sales team and once your purchase has been finalized and the license information has been updated in our CRM, you can immediately add extra PortaSIP® instances to your system in just a few minutes.

A hardware failure no longer causes a lengthy service outage if, for example, the RADIUS server you were running goes down because of a hardware failure — you can promptly move the service to a different host. This eliminates several hours of potential downtime, since there is no need for someone to travel to the collocation facility where the servers are installed. While the RADIUS is running on a different server, you have plenty of time to fix (or even replace) the defective one. By the same token, you can add new physical servers or perform maintenance on existing ones without interrupting the flow of your business, by reassigning the applications to other servers.

Also, your license key will never again be lost during the relocation now that the physical dongle has been replaced by a license file.

What is a License file?

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

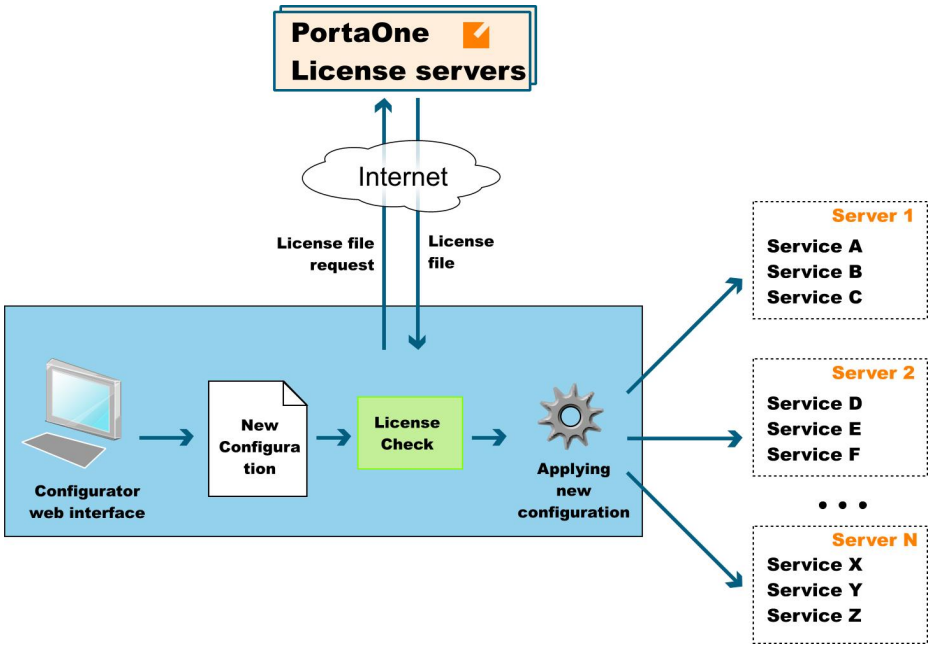
- License Instances (e.g. PortaSIP®, Billing Engine, DB, Web, RT, etc.)
- Their options (e.g. Cluster, SMP etc.)
- Information about IPs
- Expiration date
- Information about the owner
- Encryption seed and signature

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

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

How does it work?

You can make changes to your "live" system at any time (e.g. create a new RADIUS instance or move it to a different physical server) using the web interface on the Configuration Server (see Section 2 of the **PortaSwitch® Configuration Server Web Reference Guide**). When you apply the change, the Configuration Server will retrieve the **license file** from a centralized PortaOne Licensing Server and check whether all of the new configuration items (e.g. total number of RADIUS nodes 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.

2. Integration with Third-Party Systems

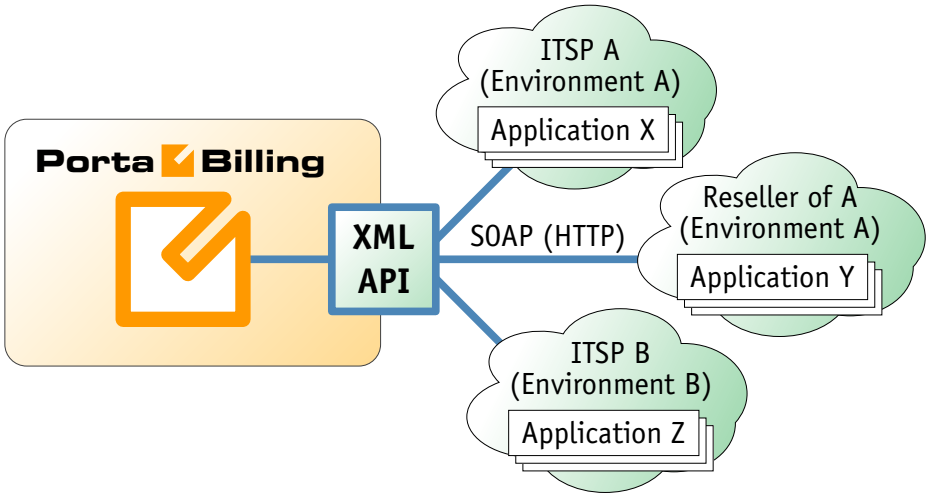
Overview

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.

XML 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 XML API. 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 and provisioning systems (where an application needs to supply a new customer's data to PortaBilling in order to activate him).

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).



This method has several advantages:

- It is based on SOAP (Simple Object Access Protocol) and HTTPS transport, so it is accessible from any platform or operating system, and all communication between the server and clients is secure.
- Since it is based on the XML and HTTP protocols, SOAP can be used in applications written in any programming language (Java,

.NET, PHP, etc.) under any OS (Unix or Windows), so that developers are free to use the tools they are most familiar with.

- The business logic embedded into the API provides integrity checks for all data modifications, as well as data composition for data retrieval.
- XML API is accessible by 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.

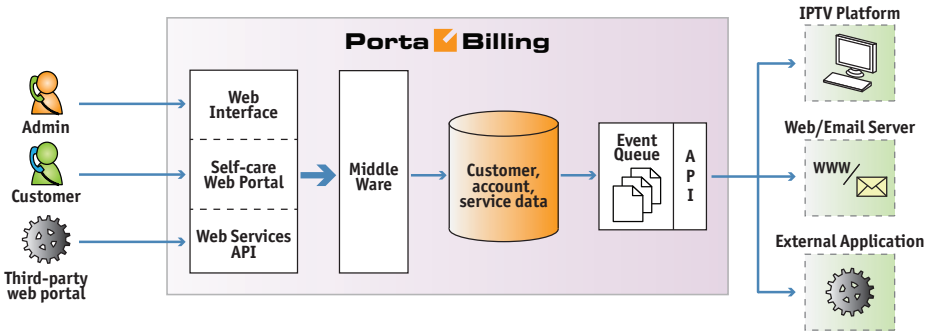
XML API allows users to perform select, update, insert or delete operations on entities such as customers or accounts. Each user has his own login credentials, and each operation he wishes to perform is analyzed to determine if it is possible with regard to general data integrity (e.g. a new account 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 API (such as available methods and data structures) are described in the **PortaSwitch® External Interfaces Guide**.

Provisioning of External Systems

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

Every modification of an object such as an account or customer in PortaBilling is recorded as an event. These events are queued in the system and then an updated service configuration for each account is pushed out to one or several provisioning plug-ins. Each of these plug-ins provides an interface for supplying data to a specific external system. This could be a text configuration file for a legacy application, or an XML API provisioning interface for a state-of-the-art service platform.



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