Orchestra NG Administrator Manual

VoiSmart

Orchestra NG Release 6.0.27

Copyright 2014 VoiSmart Srl
URL: www.voismart.com
E-mail: info@voismart.com
Phone: +39.02.70633354
Support phone: +39.02.45487888
Fax: +39.02.45487890

Notices

The information contained in this document is not designed or intended for use as critical components in human life-support systems, equipment used in hazardous environments, or nuclear control systems. VoiSmart disclaims any express or implied warranty of fitness for such uses. The information in this document is subject to change without notice. VoiSmart assumes no liability for errors that may appear in this document. Any software described in this document is furnished under license and may be used or copied only in accordance with the terms of such license.

Contents

Contents i

List of Tables iii

List of Figures v

1 Introduction 1

2 Quickstart 5

3 Basic system configuration 17

4 Orchestra NG web interface 23

5 Reports 31

6 Domains 35

7 Trunks 43

8 Routing 57

9 System 69

10 CAC 81

11 Security 89

Appendices 93

A DNS MX Records 93

B CSV Export cdr 95

C Regular Expressions 97

D Protocols and standards 101

Glossary 103

List of Tables

C.1 Regular expressions valid symbols. 97
C.2 Regular expressions examples. 98
C.3 Regular expressions substitutions. 99

List of Figures

2.1 Domains list . . . 7
2.2 Domains edit window . . . 7
2.3 Domains details window . . . 8
2.4 SIP profile edit . . . 9
2.5 Example output of TDM interfaces auto discovery . . . 10
2.6 Example of *SIP gateway* basic configuration . . . 11
2.7 Example of an automatically created carrier, from a *SIP gateway* . . . 12
2.8 Example of an TDM created carrier, with Add action . . . 12
2.9 Initial LCR creation . . . 13
2.10 Assigning LCR to a domain . . . 13
2.11 Default LCR rule . . . 14
2.12 Assigning a Carrier to a LCR . . . 14
2.13 Assigning an E.164 to a domain . . . 15

3.1 *system-config-network* application . . . 19
3.2 *system-config-network* device parameters input form . . . 19
3.3 *system-config-network* DNS setup . . . 20
3.4 Example *date* command and its output. . . . 20

4.1 Orchestra NG administrator welcome page. . . . 25
4.2 Orchestra NG administrator toolbar. . . . 25
4.3 Context menu for grid column widgets. . . . 26
4.4 Context menu for grid column widgets, visible columns selection. . . . 26
4.5 Window with two active column filters. . . . 27
4.6 Column filter for textual fields. . . . 27
4.7 Column filter for numeric fields. . . . 28
4.8 Column filter for date fields. . . . 28
4.9 Column filter for exclusive choice fields. . . . 28
4.10 Column filter for multiple selection fields. . . . 28
4.11 Grid's paging toolbar. . . . 29
4.12 Live search on the users window. . . . 29
4.13 Live search on the phone book. . . . 30

5.1 Cdrs list window. . . . 32
5.2 Cdrs export window. . . . 33

6.1 Domains list . . . 36

6.2 Domains edit window . . . 37
6.3 Domains details window . . . 37
6.4 LDAP phonebook initial configuration tab . . . 39
6.5 LDAP authentication configuration window . . . 40
6.6 Domain autoprovisiong groups creation . . . 41
6.7 Association of autoprovisiong groups with extensions . . . 41
6.8 Association between system Access Points and domain. . . . 42
6.9 Domain API access configuration . . . 42

7.1 Example output of TDM interfaces auto discovery . . . 45
7.2 Analog interface advanced parameters . . . 46
7.3 BRI interface advanced parameters . . . 47
7.4 PRI interface advanced parameters . . . 48
7.5 SIP profile basic setup . . . 51
7.6 SIP profile advanced setup . . . 52
7.7 Example of *SIP gateway* basic configuration . . . 55

8.1 Schema of LCR routing logic . . . 59
8.2 E164 numbers setup window . . . 60
8.3 E164 fax station ID setup window . . . 60
8.4 Carriers setup window . . . 61
8.5 Carriers association with trunks . . . 62
8.6 LCR setup window . . . 63
8.7 Association between LCR and tenants . . . 63
8.8 LCR rules configuration . . . 64
8.9 LCR rules test function . . . 64
8.10 LCR rules association with carriers . . . 65
8.11 Main carrier maps window. . . . 66
8.12 Carrier maps rules window. . . . 66

9.1 System settings panel . . . 70
9.2 Mailserver settings panel . . . 71
9.3 VPN panel . . . 72
9.4 Firewall configuration panel . . . 74
9.5 WiFi access points credentials setup . . . 75
9.6 Backup import . . . 76
9.7 Backup export . . . 77
9.8 Autoprovisioning configurations . . . 79
9.9 Autoprovisioning rule parameters . . . 80

10.1 CAC logical flow . . . 83
10.2 CAC Zones panel . . . 84
10.3 Example CAC Selector by network address . . . 84
10.4 Example CAC Selector by SIP domain . . . 85
10.5 CAC Limits panel . . . 85
10.6 CAC Groups panel . . . 86
10.7 CAC Association between Zones, Limits and Groups . . . 87

1

Introduction

Contents

1.1 What is Orchestra NG **2**
1.2 Overview **2**
1.3 Applications **2**
Orchestra NG with TDM circuits 2
Orchestra NG with SIP trunks 2
Orchestra NG dual homed with SIP trunks 2
Orchestra NG hybrid trunks 3

1.1 What is Orchestra NG

Orchestra NG is a full featured, multi-tenant IP-PBX that allows to offer standard PBX features over IP networks, using SIP as VoIP protocol for local extensions and hybrid TDM/SIP for interconnecting to telco networks.

1.2 Overview

An Orchestra NG system is normally composed by several items, depending on specific application deployment. In most cases there's an Orchestra NG instance running on dedicated hardware or virtual appliance, several VoIP phones compatible with the SIP protocol and one or more interconnection to PSTN circuits using SIP or TDM technology. Every component is connected to each other using standard IP networks.

1.3 Applications

Orchestra NG with TDM circuits

Orchestra NG, if installed on dedicated x86 or x86_64 servers, can be equipped with TDM cards to interconnect the system with digital or analog PSTN world. Supported interfaces are:

- Primary rate interfaces (PRI);
- Basic rate interfaces (BRI);
- Analog FXO interfaces.

Supported protocols over TDM interfaces:

- Q.931/Q.921 protocols on PRI and BRI interfaces;
- 2-wire loop start with ETSI/ITU FSK CallerID on FXO interfaces.

The local extensions are SIP ip-phones connected on the local network, connected trough an instance of the Orchestra NG SIP stack, called *SIP profile*.

TDM cards cannot be installed on virtual servers.

Orchestra NG with SIP trunks

Orchestra NG can act as a pure VoIP IP-PBX where both trunks and local extensions are SIP based. In this scenario Orchestra NG has at least two SIP stack instances, one for serving ip phones and the other one used to handle the SIP trunks, otherwise called *SIP gateway*.

Orchestra NG can connect to any VoIP provider that supports the SIP protocol, acting as a client by registering to the provider or as a server-to-server IP link.

Orchestra NG dual homed with SIP trunks

Orchestra NG supports multi-homed environments in order to be able to run different *SIP profiles* on different interfaces, separating local VoIP traffic from external one and being able to isolate public IP network from local one.

Orchestra NG hybrid trunks

As a combination of the above examples, Orchestra NG can handle simultaneously TDM and *SIP gateways*, distributing calls between the different technologies using several rules and handling inbound DIDs seamlessly, without any special handling for call coming from different interface types.

2

Quickstart

Contents

2.1 Introduction 6
2.2 Configure basic system parameters 6
2.3 Connect to the web interface 6
2.4 Configure basic domain 6
2.5 Create basic *SIP profiles* 7
2.6 Discover and configure TDM trunks 8
2.7 Create the needed *SIP gateways* 8
2.8 Create the *carriers* 9
2.9 Create the basic *LCR* 10
2.10 Associate public E.164s to your domain 13

2.1 Introduction

A basic configuration for Orchestra NG suitable for initial setup of inbound and outbound calling, is comprised of these steps:

- configure basic system parameters, see section 2.2
- connect to the web interface, see section 2.3;
- configure basic domain, see section 2.4;
- create basic *SIP profiles*, see section 2.5 on the next page;
- discover and configure TDM trunks, see section 2.6 on page 8;
- create the needed *SIP gateways*, see section 2.7 on page 8;
- create the *carriers*, see section 2.8 on page 9;
- create the basic *lcr*, see section 2.9 on page 10;
- associate public DIDs to your domain, see section 2.10 on page 13.

At the end of these tasks, please refer to User Manual quickstart to complete the domain-specific configuration.

2.2 Configure basic system parameters

A valid network configuration is needed in order to use the system. To configure it please refer to chapter 3 on page 17.

2.3 Connect to the web interface

In order to connect to the web interface, a supported browser must be used. See section 4.2 on page 24 for details.

Write into the address bar of your browser of choice the IP address of the Orchestra NG system and hit enter, like shown on figure 4.1.

Default username is *admin@example.voismart.com* with password *admin*.

2.4 Configure basic domain

By selecting the Domains button the domains list will appear, as shown on figure 2.1.

Then double click on the domain shown, or select it and edit using the Edit button. A new window will appear, as shown on figure 2.2.

Click on the Details button to configure basic Domain parameters, as shown on figure 2.3.

Edit the domain name parameters by selecting your domain name, which will be the domain part of all usernames beloging to that specific company. It will be also the SIP domain used in configuring phones, for SIP authentication. An IP address can be used as domain name, which helps in environments where no DNS or SIP SRV records are used.

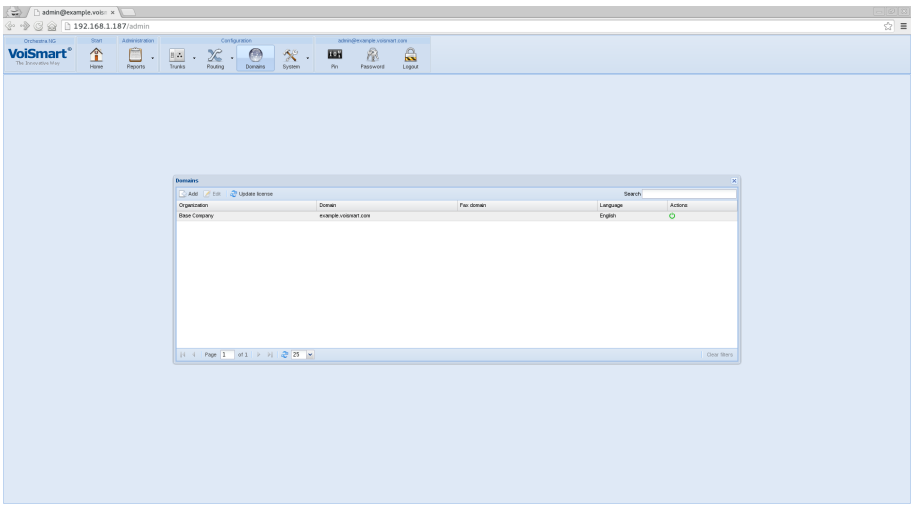

Figure 2.1: Domains list

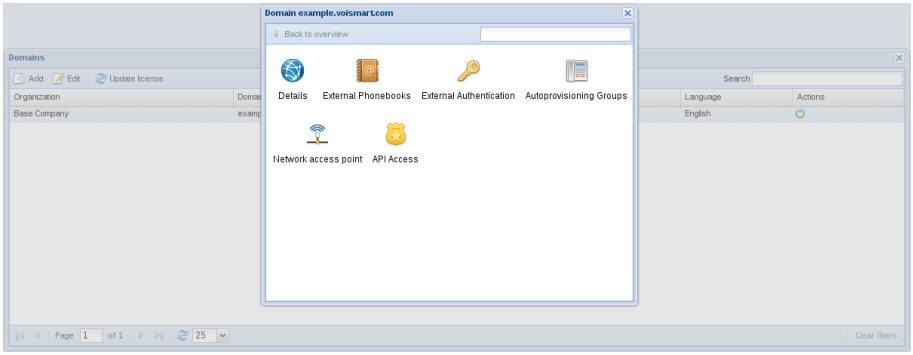

Figure 2.2: Domains edit window

Setup also various objects quantities, based on your license. For example, if your license has 10 Users, you can assign up to 10 users to the specific domain. The domain administrator will be able to setup up the number of configured users here. The remaining users can be assigned, for example, to a second domain. If more items than the currently available are assigned, an error will pop up when saving.

Press Save when done.

2.5 Create basic *SIP profiles*

Select Trunks → *SIP profiles* to open the *SIP profiles* configuration window. Select Add to add a new one. A new window will appear, as shown on figure 2.4.

Fill up *SIP profile* name as *internal*, configure SIP port as 5060, select at least one codec from the list at the bottom of the window and enable the flag *Auth calls*.

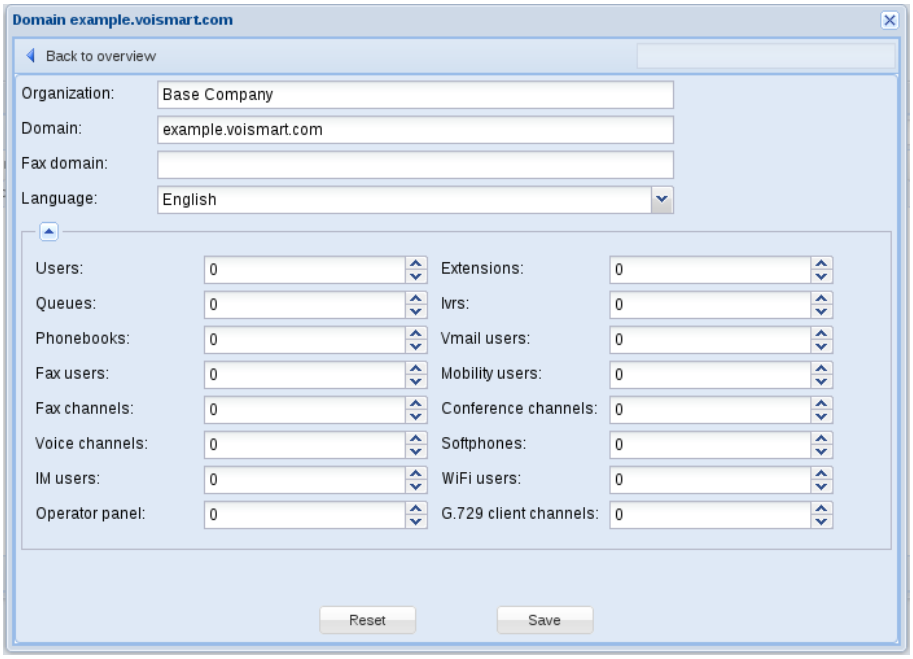

Figure 2.3: Domains details window

Press Save when done and wait for confirmation pop up. This will be the profile used by local extensions.

If an external *SIP gateway* is needed, create another profile by calling it *external*, assign port 5080 and be sure to *not* enable the flag *Auth calls*. This will be the profile used for external VoIP calls.

Please refer to section 7.3 on page 48 for advanced configuration parameters.

2.6 Discover and configure TDM trunks

If TDM interfaces are installed in the system, they need to be configured. If no interfaces are present, this section can be skipped. Orchestra NG supports TDM interfaces auto discovery, in order to simplify configuration. Select Trunks → *TDM Interfaces* to open the *TDM Interfaces* window and select *Discovery*. At the end of the discovery process a pop up will show the completion of the operation. A list of available interfaces are shown. Please see figure 2.5 for an example.

Configure *TDM Interfaces* roles and advanced parameters as needed. Please refer to section 7.2 on page 44 for further details.

2.7 Create the needed *SIP gateways*

If interconnection with SIP trunks is needed, they can be configured from Trunks → *SIP gateways* menu. Select *Add* to add a new *SIP gateway*, fill up the Name field with provider sip server domain, select on which *SIP profile* the *SIP gateway* must run (for example external), add username and password

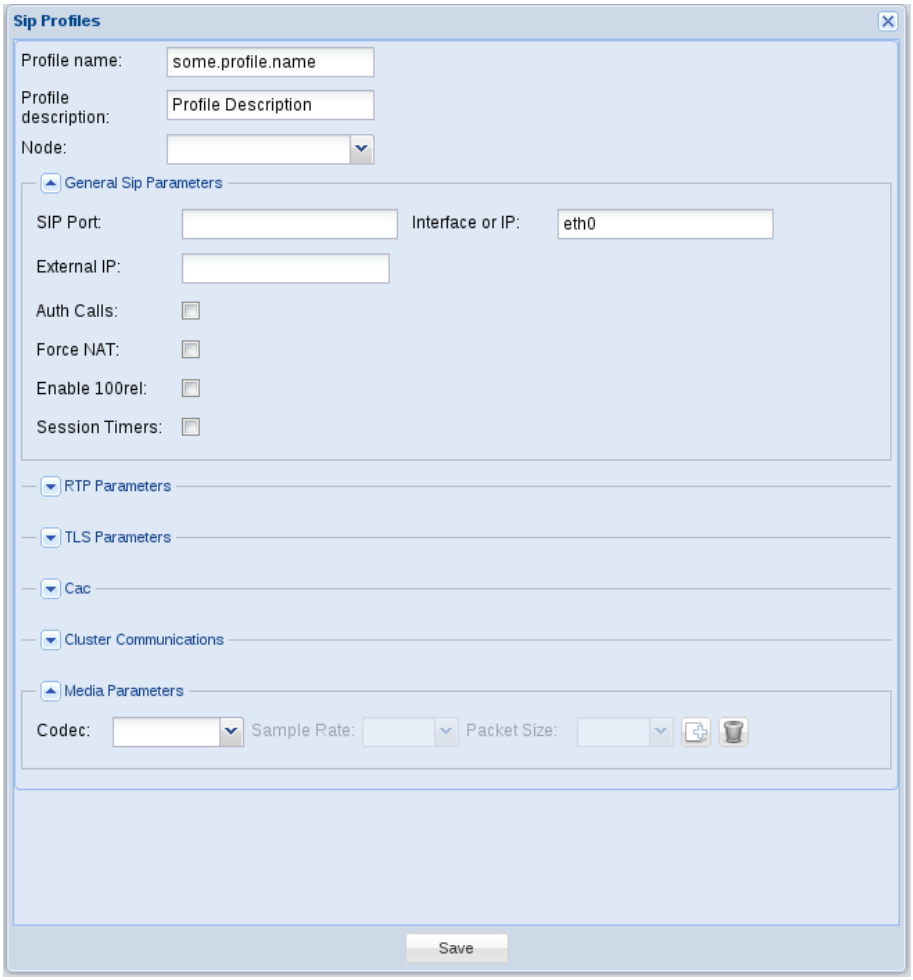

Figure 2.4: SIP profile edit

if needed. Enable the register flag if registration to the provider sip server is required. Please refer to section 7.4 on page 50 for further details or advanced parameters.

A basic configuration is shown on figure 2.6.

Press Save when done and wait for confirmation pop up.

2.8 Create the *carriers*

Carriers are used to associate one or more trunks to a single object that can be used in LCR rules. If more than one trunk is associated to a single carrier, calls will be dispatched to the carrier in a random fashion, falling back to the next one if the nth fails.

Carriers configuration is accessible from the Routing → Carriers menu.

When a *SIP gateway* is created, a related carrier will be automatically created in disabled state, as shown in figure 2.7 on page 12. The carrier just

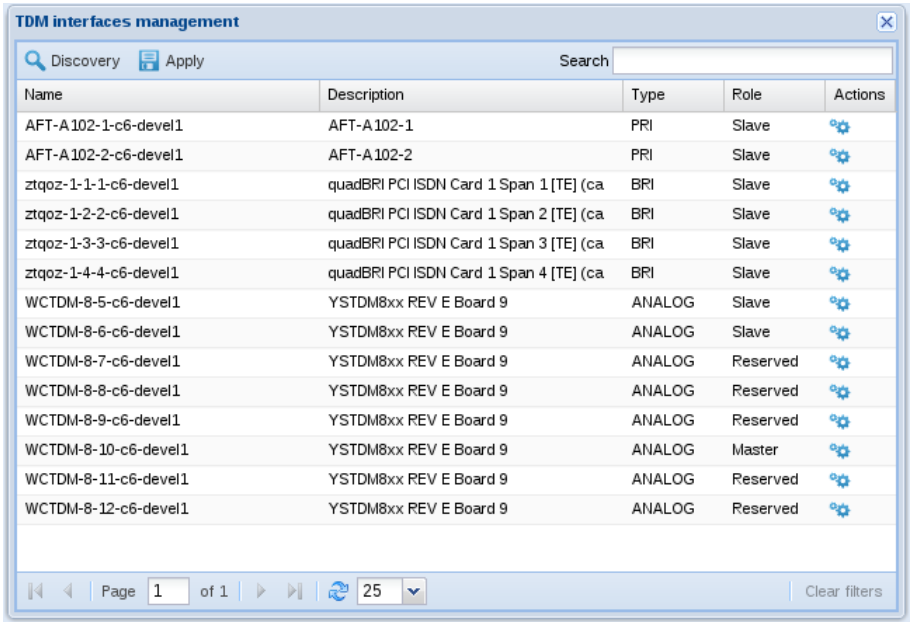

Figure 2.5: Example output of TDM interfaces auto discovery

needs to be enabled by clicking on the Enable/Disable action button.

If TDM interfaces are used, a new carrier needs to be created by using the Add button, filling in required details and confirming with Update and enabling it. Now link the just created carrier with the needed trunk by selecting the Associate Trunks action and selecting with Add, like shown in figure 2.8 on page 12.

Please refer to section 8.4 on page 58 for further details or advanced parameters.

2.9 Create the basic *LCR*

Least-Cost Router, or LCR, is an object that allows to select which trunk to use when placing an external call, using several rules that match the dialed number. To create a new LCR go to the Routing → LCR menu.

Use *Add* button to create a new LCR, like shown on figure 2.9 on page 13. After confirming with *Update*, assign at least one domain to it, using the globe action icon on the right of the window.

Now assign a domain, like shown on figure 2.10 on page 13. When done, close the domain assignment window and press next to go to LCR Rules tab. A default match all rule is automatically created, like shown on figure 2.11. Select the rule and go on with *Next* button to last step, carrier assignment. Assign at least one carrier to the rule, as shown on figure 2.12. Multiple carriers can be added and will be used in the displayed priority. Priority can be modified dragging the entry and dropping it in th desired position. Press Save when done.

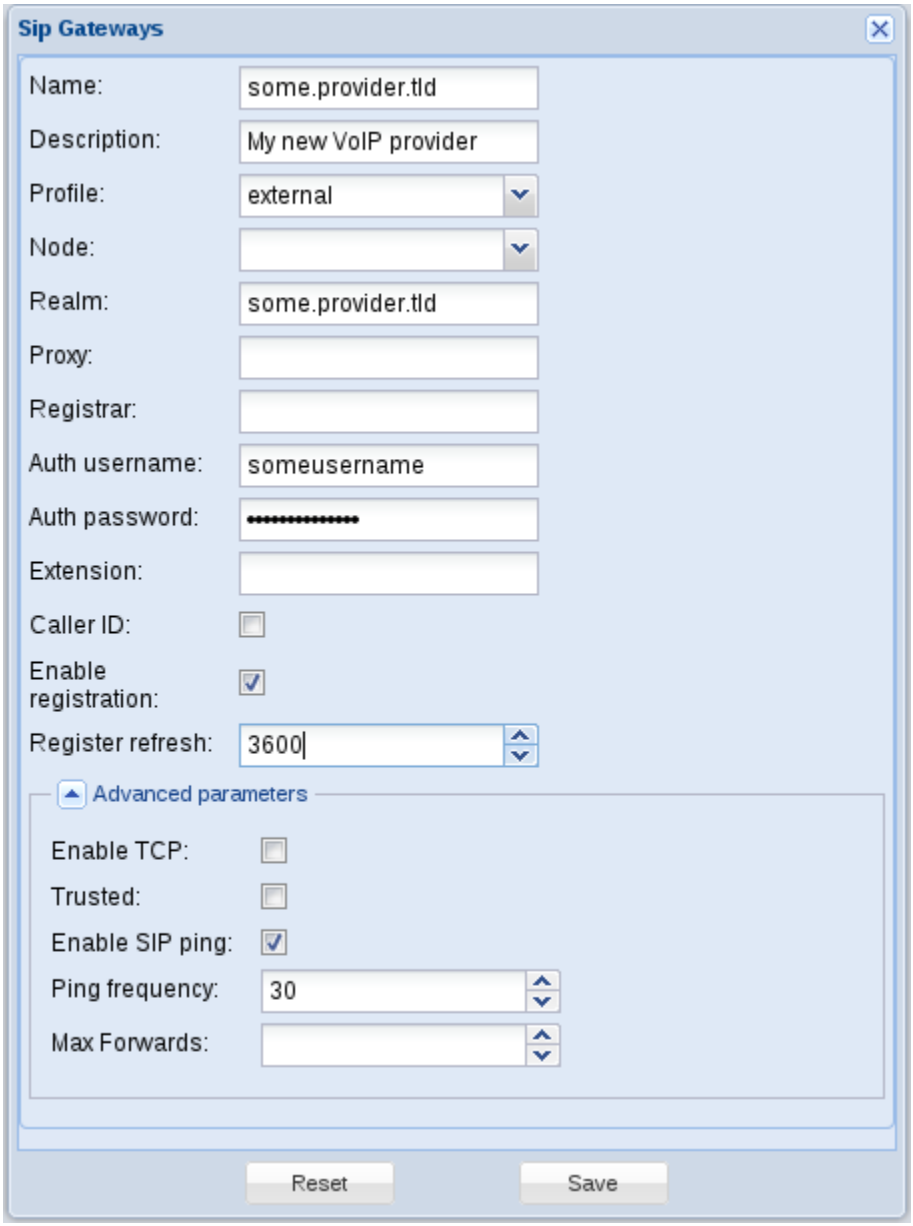

Figure 2.6: Example of *SIP gateway* basic configuration

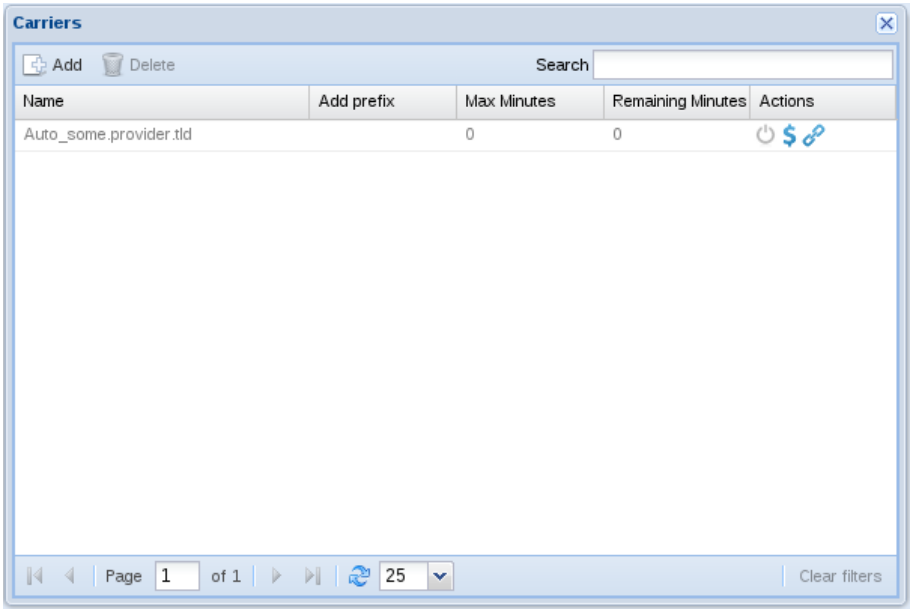

Figure 2.7: Example of an automatically created carrier, from a *SIP gateway*

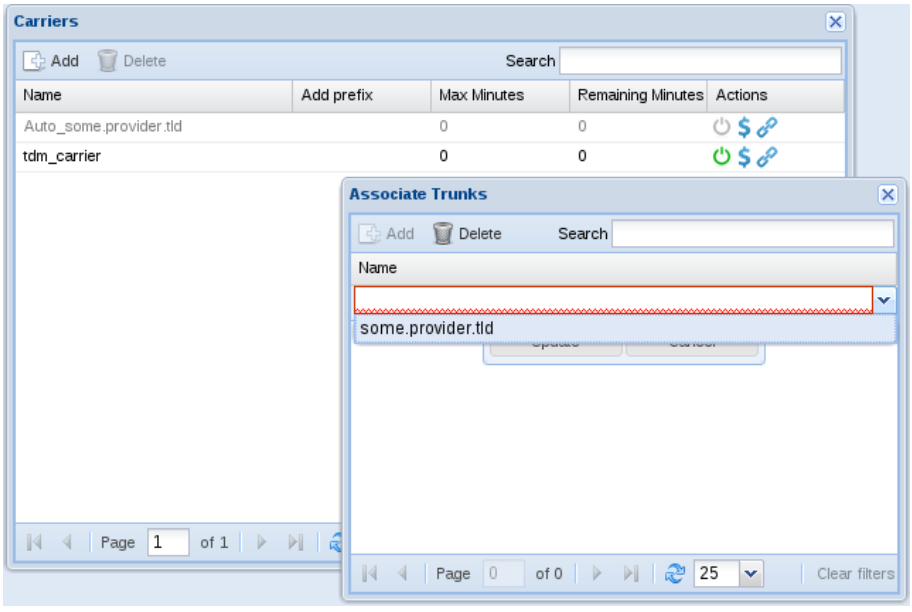

Figure 2.8: Example of an TDM created carrier, with Add action

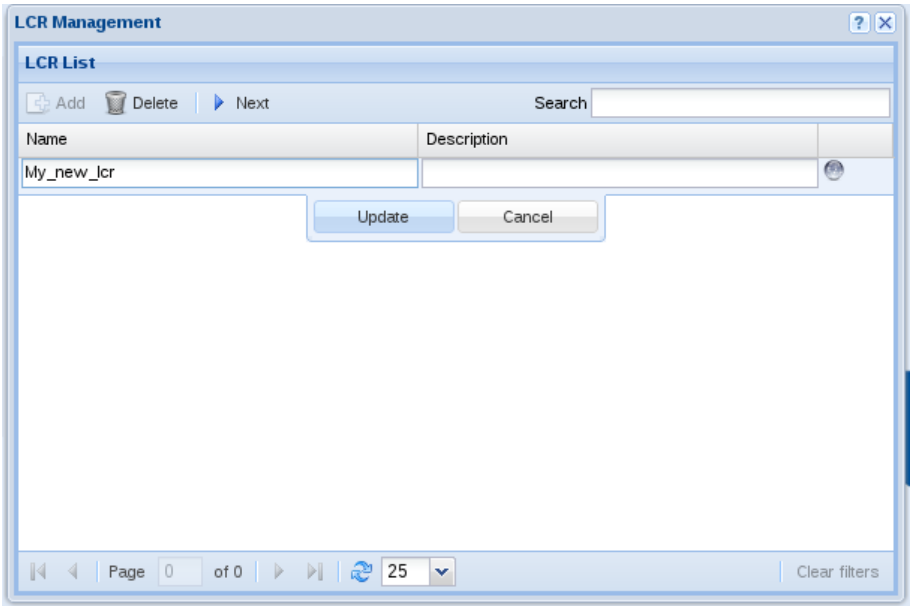

Figure 2.9: Initial LCR creation

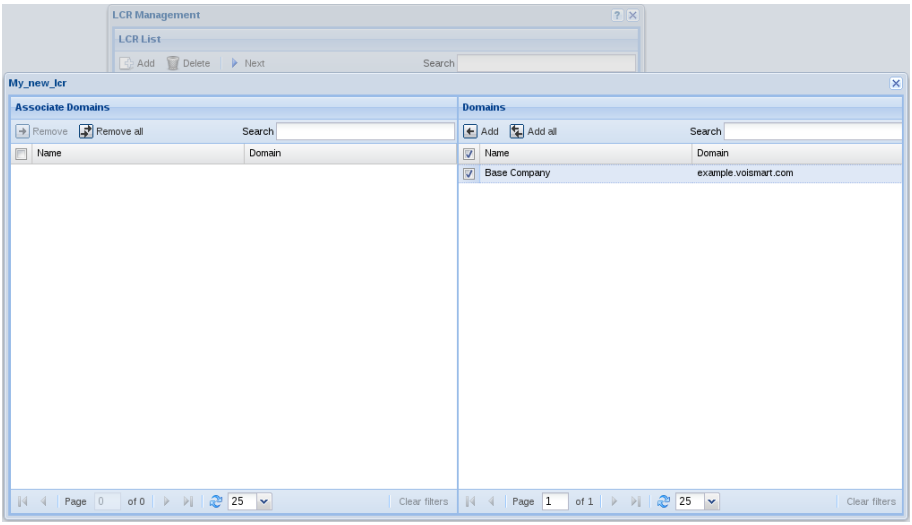

Figure 2.10: Assigning LCR to a domain

Please refer to section 8.5 on page 61 for further details or advanced parameters.

2.10 Associate public E.164s to your domain

To allow inbound calls, inbound E.164s must be configured and assigned to each domain, from the Routing → Inbound E.164s menu. E.164 entries are

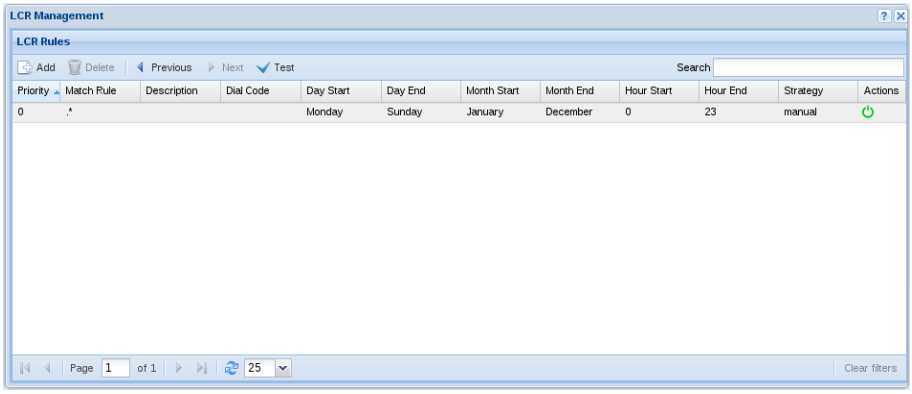

Figure 2.11: Default LCR rule

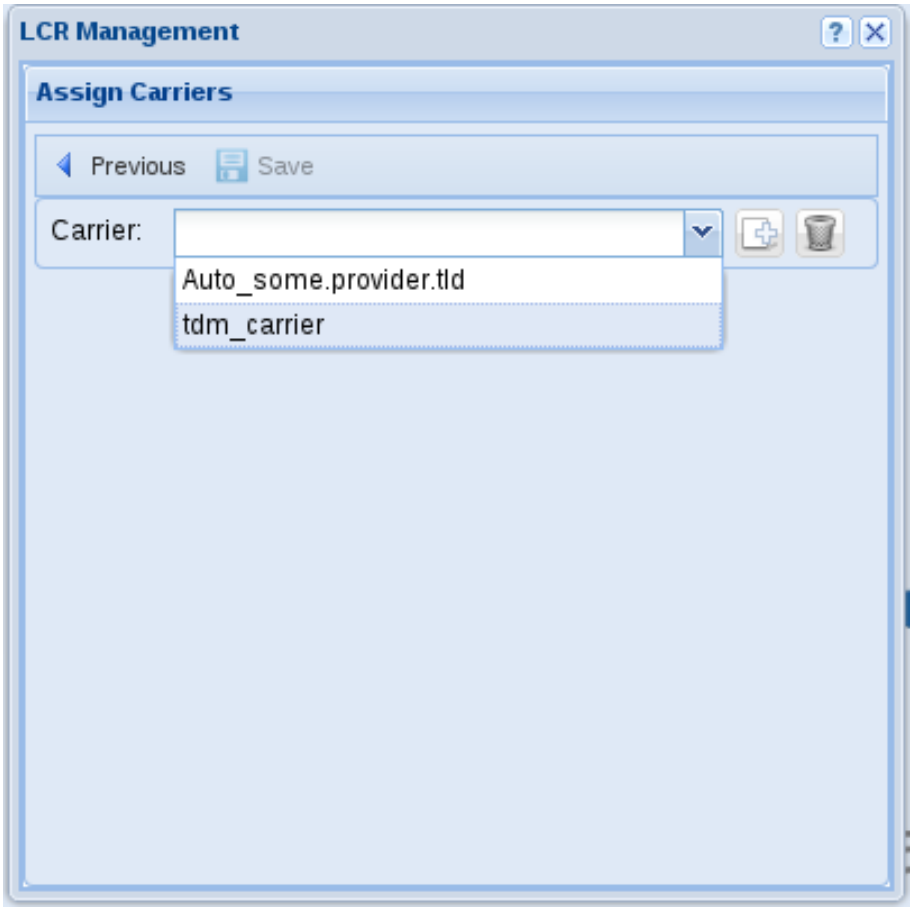

Figure 2.12: Assigning a Carrier to a LCR

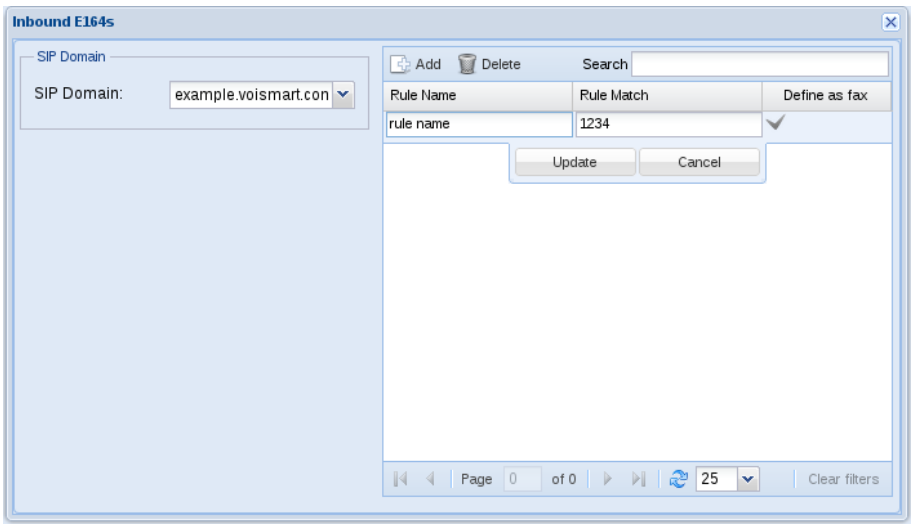

Figure 2.13: Assigning an E.164 to a domain

used to direct calls to their own domain for further, domain-specific, processing. From the E.164s windows select the domain, and Add a new rule. Rules can be a single number or a match, using regular expression syntax. See an example on figure 2.13.

Please refer to section 8.3 on page 58 for further details or advanced parameters.

3

Basic system configuration

Contents

3.1 Introduction **18**
3.2 Network **18**
3.3 Time and date **18**
3.4 Software upgrades **21**

3.1 Introduction

Orchestra NG is a Linux-based system, currently using CentOS 6 as distribution.

In the current version basic system configuration must be done using the Linux shell, so a minimal Linux knowledge is required to setup the system.

To connect to the system an ssh client is needed. For the Windows platform PuTTY is the most common client, for other platforms a variety on free clients exists.

The following operations must be done by ssh'ing into the Orchestra NG system if IP address is known, otherwise using local shell.

All operations must be done as root user. The root password depends on the installation type:

- if the system is an appliance the root password is sent along the appliance itself;
- if the system is a virtual image, the root password is *123456*;
- if the system has been installed from scratch, the root password is already known.

3.2 Network

The network configuration is done by invoking the system tool *system-config-network*. Start with Device configuration and select the device that needs to be configured. Fill up the form with needed parameters, press OK to confirm and Save. Then move to DNS configuration to setup DNS parameters. Press OK to confirm and save everything by selecting Save&Quit.

Refer to figures 3.1 on the next page, 3.2 on the facing page and 3.3 on page 20 for sample outputs.

Now apply the network configuration by issuing the command *service network restart*.

> ☞ If the IP address is changed, all network connections to the system will be lost, including the remote secure shell.

> ☞ A valid DNS and gateway configuration are necessary for the proper functioning of the system.

3.3 Time and date

On the Orchestra NG an ntp client is already installed, so the time is kept automatically in sync using the Network Time Protocol. If the system is disconnected from the internet or an initial setup is needed, time and date setup from the shell is possible. To change the date and time, issue the command *date MMDDhhmmYYYY*

- MM: two digit month number;
- DD: two digit date;

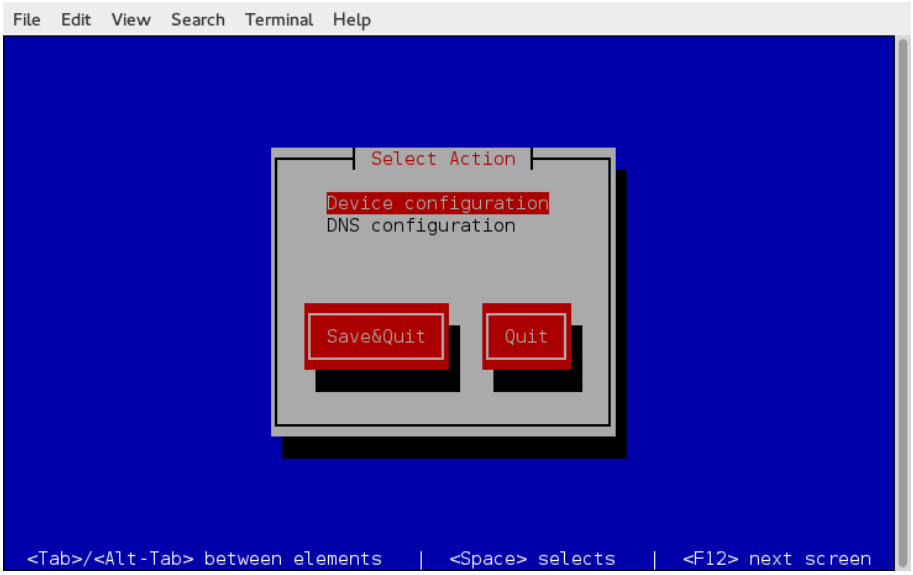

Figure 3.1: *system-config-network* application

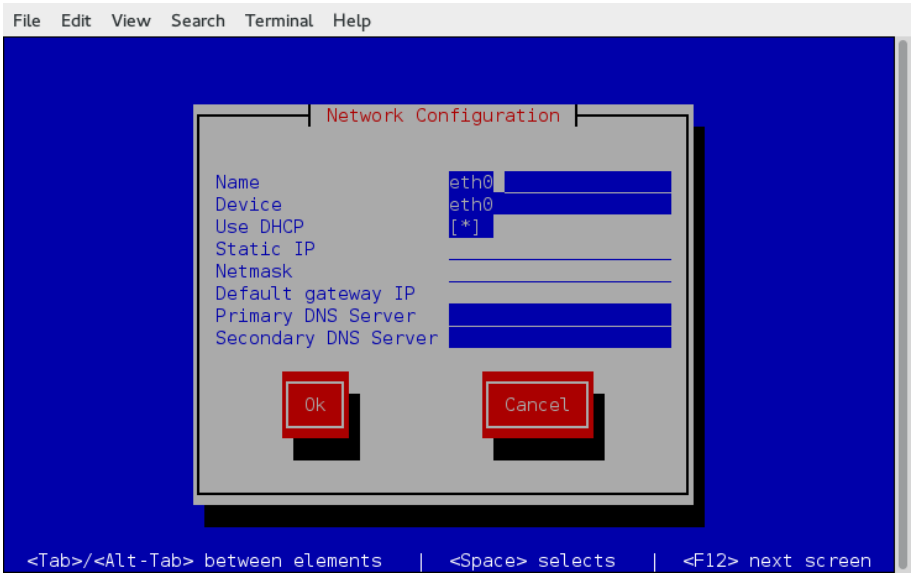

Figure 3.2: *system-config-network* device parameters input form

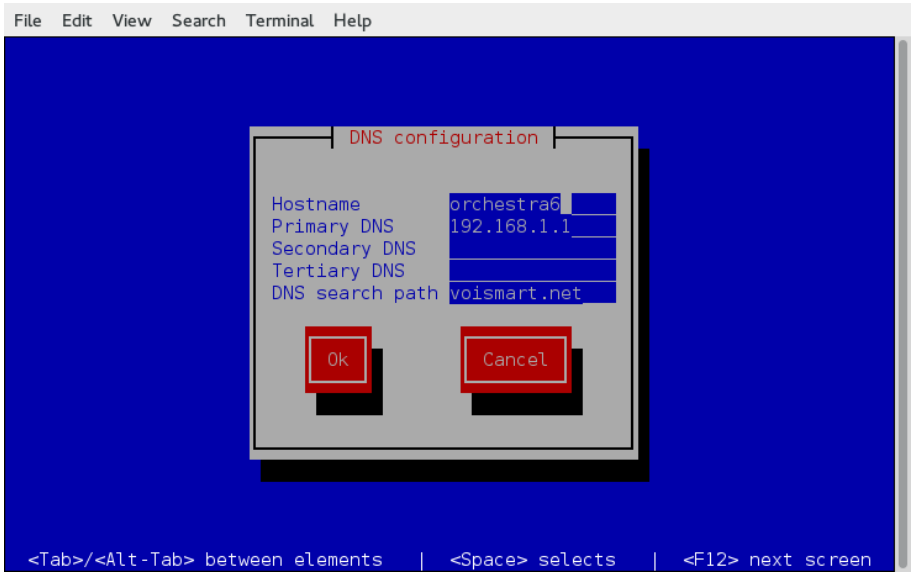

Figure 3.3: *system-config-network* DNS setup

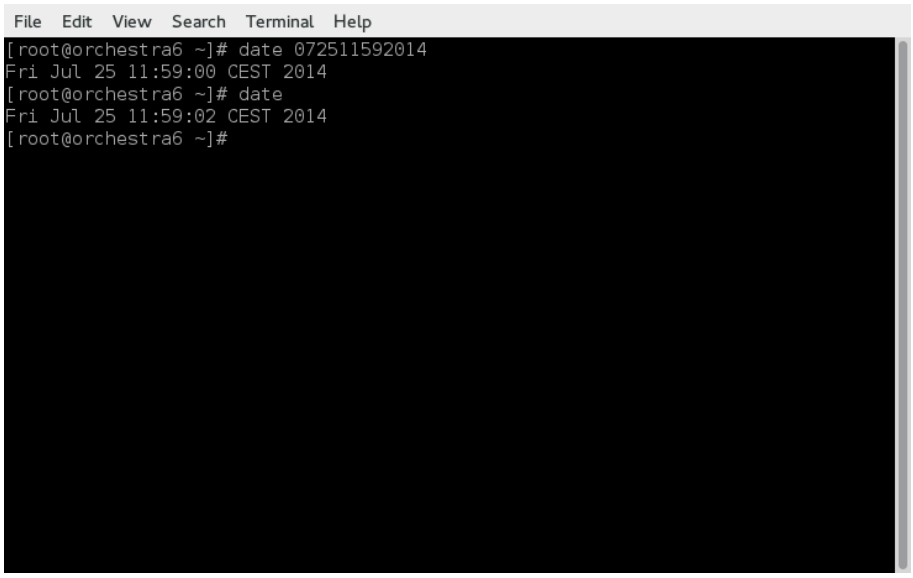

Figure 3.4: Example *date* command and its output.

- hh: two digit hour (24 hour system);
- mm: two digit minute;
- YYYY: four digit of year.

Refer to figure 3.4 for example command and its output.

3.4 Software upgrades

Orchestra NG uses the RPM packaging system to deliver software and yum to manage installed software.

To update the installed system only two commands are needed to be run as root:

- *yum clean all* to clean up the yum cache;
- *yum update* to run the update;

Yum will search for updates and prompt for confirmation before downloading and applying the updates. Press Y to confirm or N to abort the upgrade.

> ☞ A valid DNS, gateway configuration, internet access and no web proxies between are necessary conditions for a correct system update.

4

Orchestra NG web interface

Contents

4.1 Introduction **24**
4.2 Supported Browsers **24**
4.3 Access to the web interface **24**
4.4 Main toolbar **24**
4.5 Common GUI elements **25**
Grid column menu 25
Grid's paging toolbar 29
Live search feature 29

4.1 Introduction

Orchestra NG is web based system, where all services are used and managed using an HTML5 compliant web interface. There are two web GUIs, one for the domain or tenant, the other one for platform configurations that are on top of domains.

4.2 Supported Browsers

Because of leveraging on advanced JavaScript/CSS techniques, some restrictions on supported browser are applied, in order to guarantee the best user experience. Currently supported browsers are:

- Google Chrome, from version 24;
- Mozilla Firefox, from version 30;
- Microsoft Internet Explorer, from version 11.

4.3 Access to the web interface

Orchestra NG has some basic params, like IP address or login details.

- default username: *admin@example.voismart.com*;
- default password: *admin*;
- default ip address if installed as software: *DHCP*, unless differently specified during operating system installation;
- default ip address if bought as appliance: *192.168.0.250*;
- domain web interface url: `http://<ip address>`;
- system administrator web interface url: `http://<ip address>/admin`.

> ✍ <default ip address> means the currently configured IP address of the Orchestra NG installation.

On figure 4.1 on the next page a sample login interface is shown.

4.4 Main toolbar

The main menu, as shown on figure 4.2 on the facing page, is a set of submenus used to access all global platform configurations. A summary of all menus follows:

- Orchestra NG: when clicked displays the build version;
- Home: not yet used;
- Reports: access to CDR and FDR by selecting the corresponding menu options, see chapter 5 on page 31;

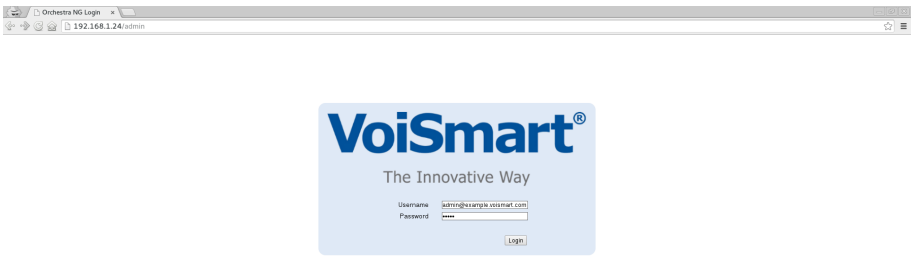

Figure 4.1: Orchestra NG administrator welcome page.

Figure 4.2: Orchestra NG administrator toolbar.

- Trunks: access to *SIP profiles*, SIP trunks and TDM trunks configuration, see chapter 7;
- Routing: access to Inbound E.164, carriers and LCRs configuration, see chapter 8;
- Domains: access to domains configuration, see chapter 6;
- System: access to various system related functions, like SMTP, VPN for remote support, CAC and Wi-Fi, see chapter 9;
- Pin: see User manual for further informations;
- Password: allows to change the logged-in user password;
- Logout: terminate the current session, redirecting to the login page.

4.5 Common GUI elements

This section describes some common elements and functions that are used extensively throughout the Orchestra NG web interface.

Grid column menu

The grid interface is commonly used to show a list of records with a varying level of details and functions. One common function is the possibility to filter the records according to some search criteria and to hide some of the fields.

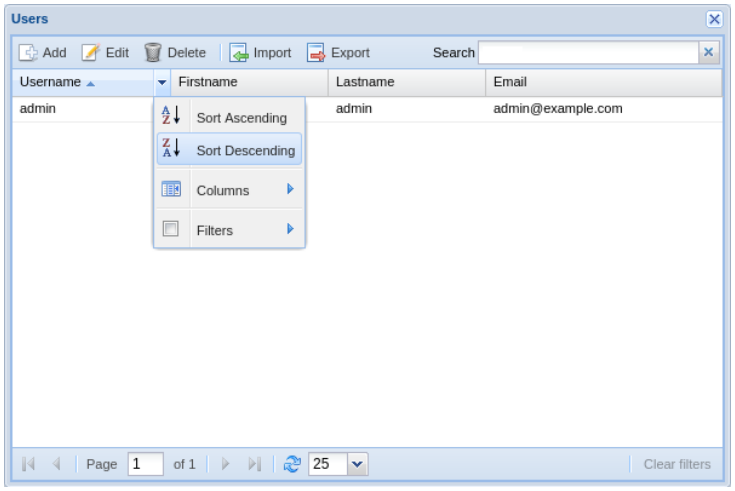

Figure 4.3: Context menu for grid column widgets.

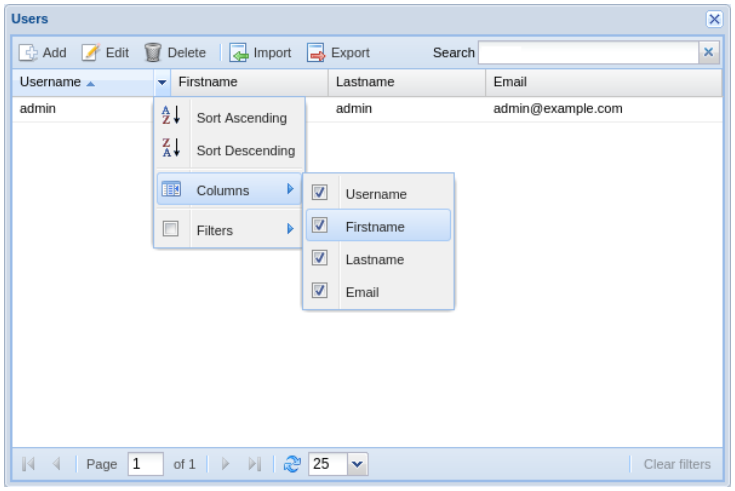

Figure 4.4: Context menu for grid column widgets, visible columns selection.

These operations are possible by opening the context menu shown in figure 4.3 by clicking the small triangle on any of the column headers.

The first two items in that menu sort the records in ascending or descending order with respect to that column. It is also possible to toggle the sort order by simply clicking on the column label repeatedly.

The columns submenu (figure 4.4), can be used to show and hide the visible columns.

The last submenu is used to filter records according to the chosen field. When a filter is active on one or more column, the filtered columns are shown in a different font, and a message is shown in the bottom-right corner of the window to warn the user that only a subset of all the records is currently being displayed. For example, in figure 4.5, there are two filters active on the fields

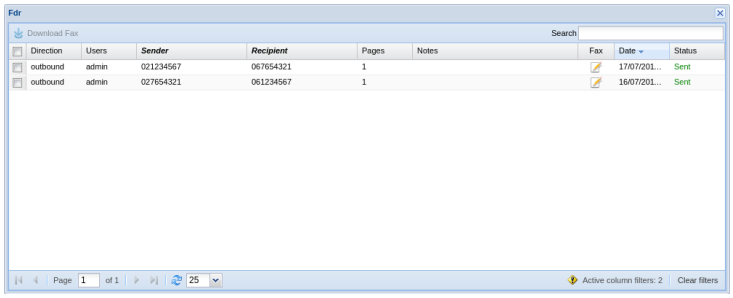

Figure 4.5: Window with two active column filters.

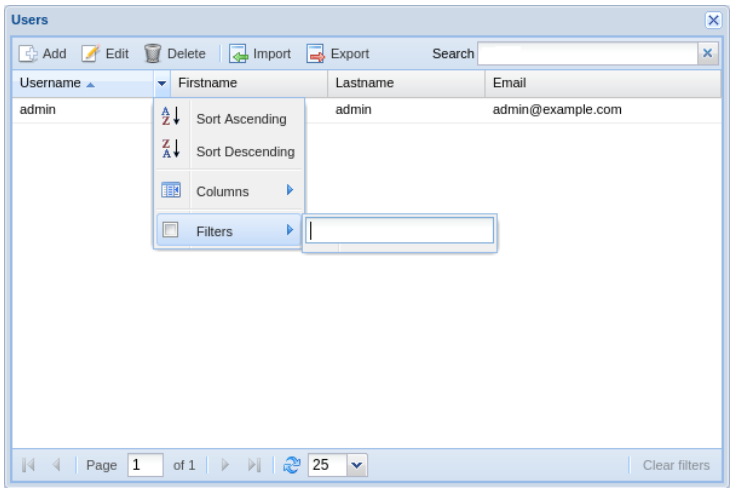

Figure 4.6: Column filter for textual fields.

"Sender" and "Recipient".

The application of a column filter is slightly different for different kind of fields:

Textual fields: simply enter the desired text to filter, figure 4.6;

Numerical fields: there are three input fields available (figure 4.7), to filter for records with values greater than, less than or strictly equal to the desired value;

Date fields: there are three date pickers to filter for records created before, after or on the selected date, figure 4.8;

Fixed choice: here, the filtering is done by selecting among the available choices. The selection can be exclusive (only one possible value can be selected, as in figure 4.9), or multiple (more than one values can be selected, as in figure 4.10).

To remove a column filter, simply deselect them from the same context menu used to create it, or just press the "Clear filters" button on the lower-right corner of the window (see figure 4.5).

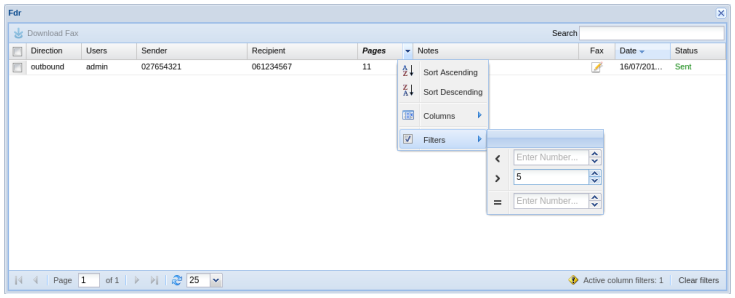

Figure 4.7: Column filter for numeric fields.

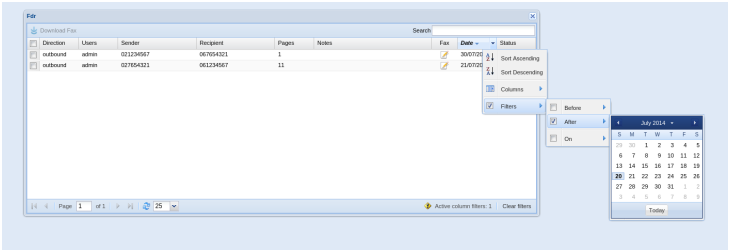

Figure 4.8: Column filter for date fields.

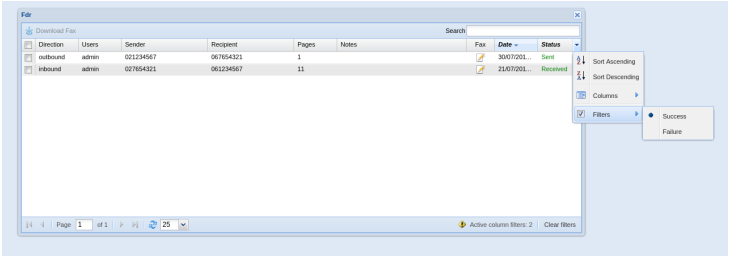

Figure 4.9: Column filter for exclusive choice fields.

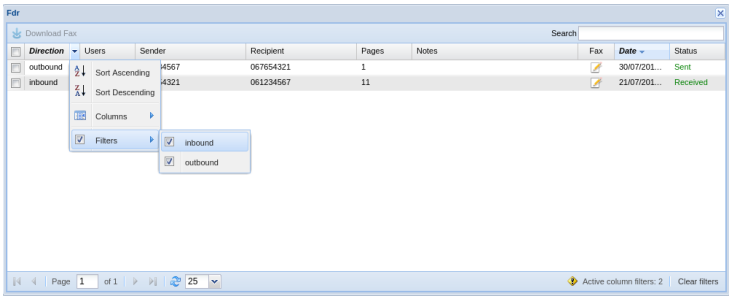

Figure 4.10: Column filter for multiple selection fields.

Figure 4.11: Grid's paging toolbar.

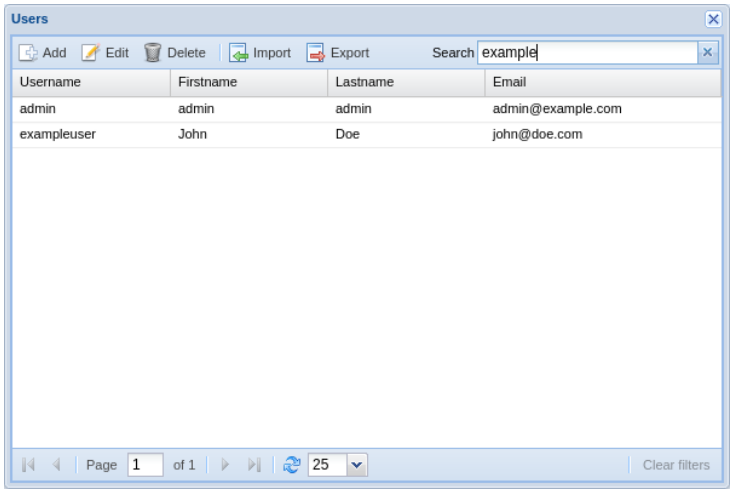

Figure 4.12: Live search on the users window.

Grid's paging toolbar

Every grid dialog, has a paging toolbar to browse through grid's items (figure 4.11).

All of the items currently visible in a grid are collectively called a page, and the paging toolbar's main purpose is to browse through pages, using the little arrows which allow to move to the first, previous, next and last page. The currently displayed page is shown in the center along with the total number of pages.

The small curved double arrow, reloads grid's items, and the small selector near it, modifies the page size, i.e. the number of items which form a page.

The last button, clears the currently active column filters, see section 4.5 on page 25 for details.

Live search feature

This feature is available in many of the dialog windows in the Orchestra NG interface. Searching content is a pervasive feature, and by entering text in a live search field, the content will be filtered in a sensible way dependent to the context.

For example, figures 4.12 and 4.13 show an example of searching through system's users and phone book contacts in a coherent way, even if the underlying search details are different.

To clear the search, just click on the small button at the right end of the input field.

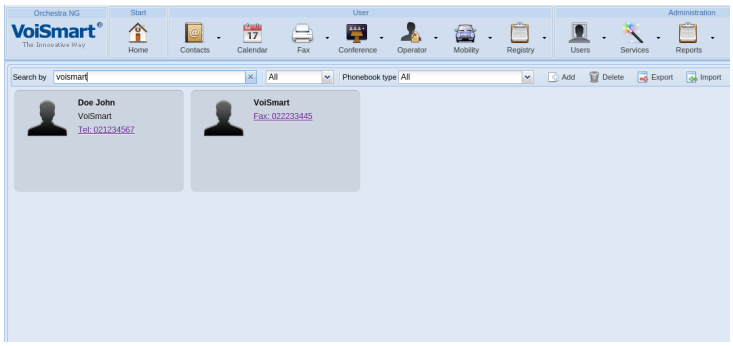

Figure 4.13: Live search on the phone book.

5

Reports

Contents

5.1 Reports **32**
Cdrs 32
Fdrs 33

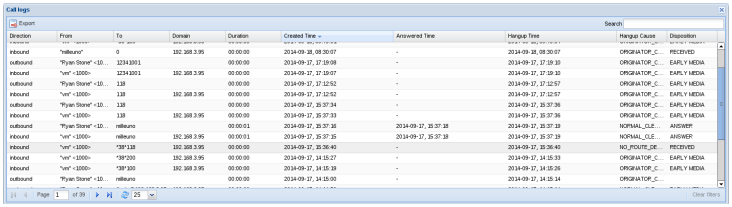

Figure 5.1: Cdrs list window.

5.1 Reports

This section shows how calls and faxes, in system-wide, are stored by Orchestra NG.

Cdrs

Orchestra NG stores a complete log of all inbound and outbound calls for logging purposes. System-wide call logs can be accessed by clicking on Reports → Call logs and shows all call records originated or received from users in all configurated domains.

The Call logs dialog looks like the one in figure 5.1 and consists in a list of call records with the following fields:

Direction: call direction, can only be "inbound" or "outbound". "Inbound" direction means that call is directed to pbx, while "outbound" direction means that call is generated by pbx;

From: the sender's caller id name;

To: the recipient's number;

Domain: (only available to the domain administrator), show call's domain;

Duration: the call's duration;

Created Time: date and time when the channel was created;

Answered Time: date and time when the channel was answered;

Hangup Time: date and time when the channel was closed;

Hangup Cause: hangup's cause;

Disposition: when available, it is hangup cause returned.

> ✍ The number of records for each call may vary depending on which service is called.

You can also export all or in part call records as a CSV file. In dialog shown in figure 5.1 by clicking on button, a new window will open as shown in figure 5.2.

In this dialog you can choose to export all records or to export records in a temporal range. In system administration web interface you can also filter

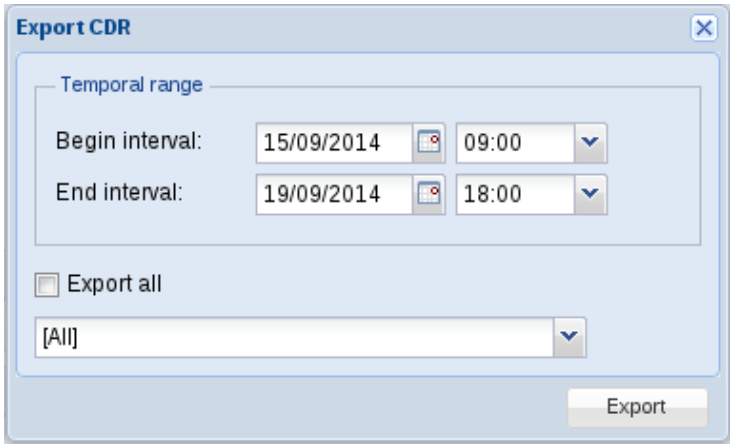

Figure 5.2: Cdrs export window.

records by selecting domain using proper check box. By clicking on *Export* button export process will start and when it will be finished, you will be able to download a CSV file. An example of CSV file is shown in the appendix B on page 95.

✍ Fields and value exported in CSV file are not the same shown in GUI (i.e. date values are exported as Unix Timestamp microseconds).

☞ If there are many call records, the export process can take a long time, so please wait!

Fdrs

Orchestra NG stores a complete log of all inbound and outbound faxes for logging purposes. System-wide fax logs can be accessed by clicking on Reports → Fax logs and shows all fax records originated or received from users in all configurated domains. For more details about fdrs consult user manual.

6

Domains

Contents

6.1 Introduction **36**
6.2 Concepts **36**
6.3 *Details* **38**
6.4 *External Phonebooks* **38**
Ldap Params 38
Ldap Mappings 39
6.5 *External Authentication* **40**
6.6 *Autoprovisioning Groups* **40**
6.7 *Network access point* **41**
6.8 *API Access* **41**

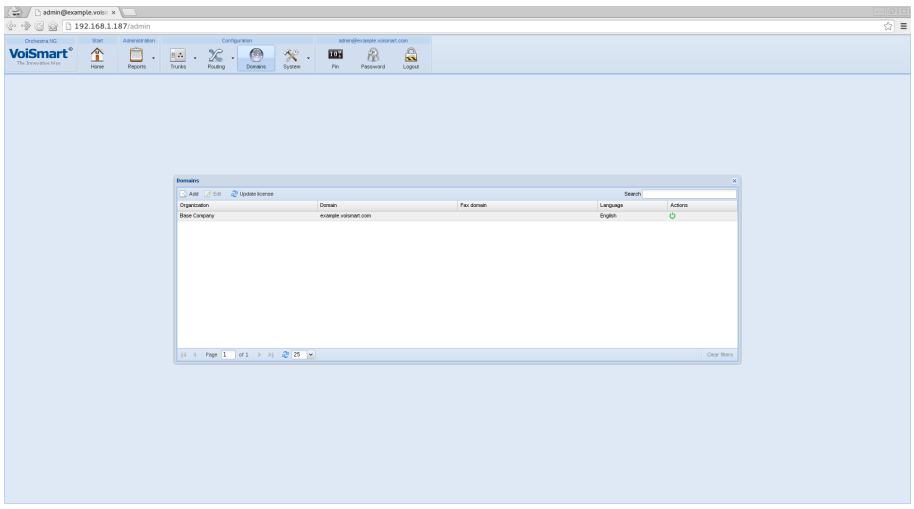

Figure 6.1: Domains list

6.1 Introduction

The Domains permits to configure each tenant parameter and to assign a subset of license keys to it.

> This feature is licensed, you can create or enable items only if you have purchased a domains license.

6.2 Concepts

Domains permit to create several, totally isolated between themselves, fully featured pbx instances inside a single Orchestra NG system. A domain can have its own trunks, enabled features and remote connections with LDAP for authentication and directory service.

The domain name is also the SIP domain used for endpoint authentication and registration.

To edit a domain select the Domains button, which will bring up a list of currently configured domains as shown on figure 6.1.

> A domain cannot be deleted once configured, but can be disabled by clicking on the corresponding action on the domains list window. See figure 6.1.

> When a domain name is changed, various services will be restarted, as the confirmation box alerts. Also the current internal phonebook for the specific domain is lost and registered extensions will fail to register unless reconfigured for the new domain.

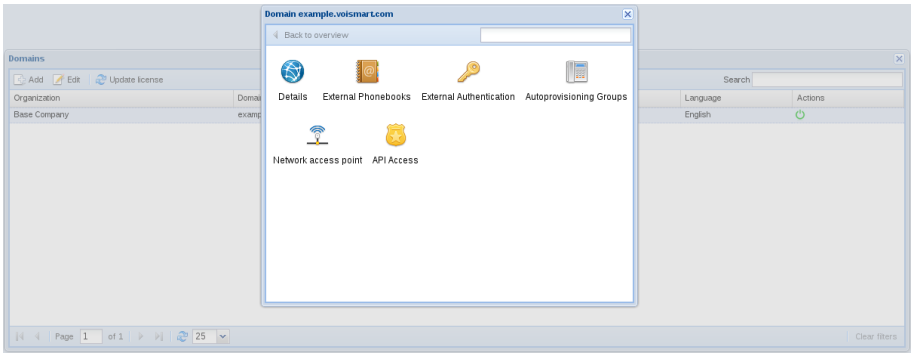

Figure 6.2: Domains edit window

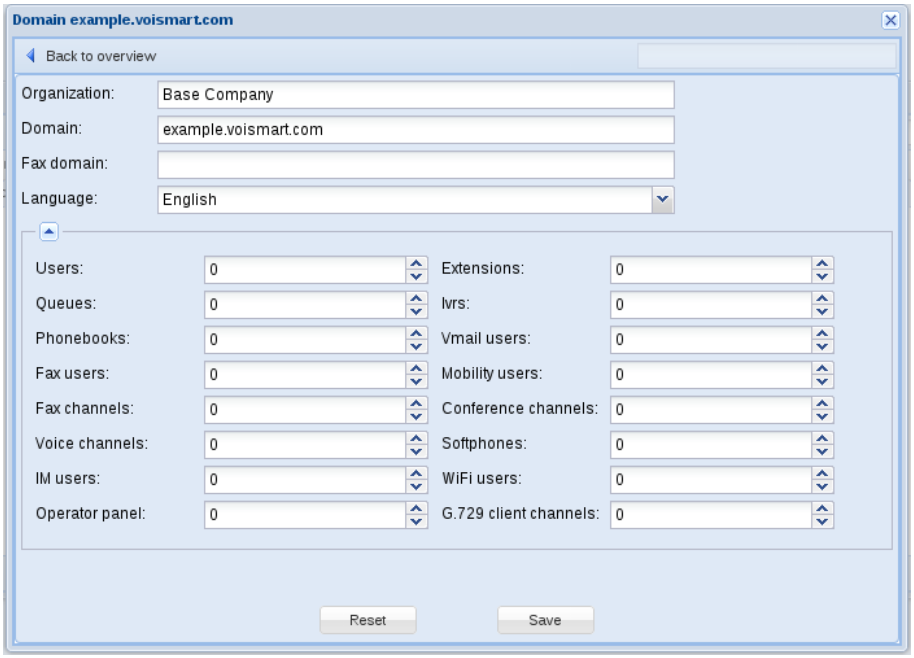

Figure 6.3: Domains details window

6.3 *Details*

The *Details* icon allows to setup domain basic parameters and the corresponding FQDN.

Organization A short, descriptive name of the company holding the specific domain.

Domain The FQDN name of the domain. It represents the user part of login URIs and the SIP domain and realm used for authentication.

Fax domain A FQDN used for email to fax gateway functions. Emails sent to this *Fax domain* will get routed to the corresponding tenant and processed by fax server service. Refer also to appendix A for a quick DNS walkthrough for email services.

Language The default language for the domain. Can be overridden on a per user basis.

License parameters Orchestra NG license comprises several global parameters, that can be assigned to each tenant in order to allow them only a subset of the globally allowed features.

All the other parameters represent quantities of *objects* that can be assigned to the tenant and will get subtracted from the global license counter.

> If the license has 10 usable users, you can assign all 10 to one tenant, or split them between several ones, for example 3 on the first, 5 on the second and 2 on the third. This concept can be applied on all other numeric paramenters.

6.4 *External Phonebooks*

> This feature is licensed, you can create or enable items only if you have purchased a phonebooks license.

By default each domain has an internal phonebook storage defined, which can be used without any further configurations. If connection to an external phonebook is needed, is possible from this section to configure one or more external LDAP server. To add a new server, just select the *Add* button and fill the parameters, as instructed below. *Name* is a label assigned to each configuration. See figure 6.4 on the facing page for reference.

Ldap Params

This section setups connection to the LDAP server and server tuning.

Ldap Server hostname or IP address of the remote server;

Ldap Port remote server port;

Ldap User username for binding to the server;

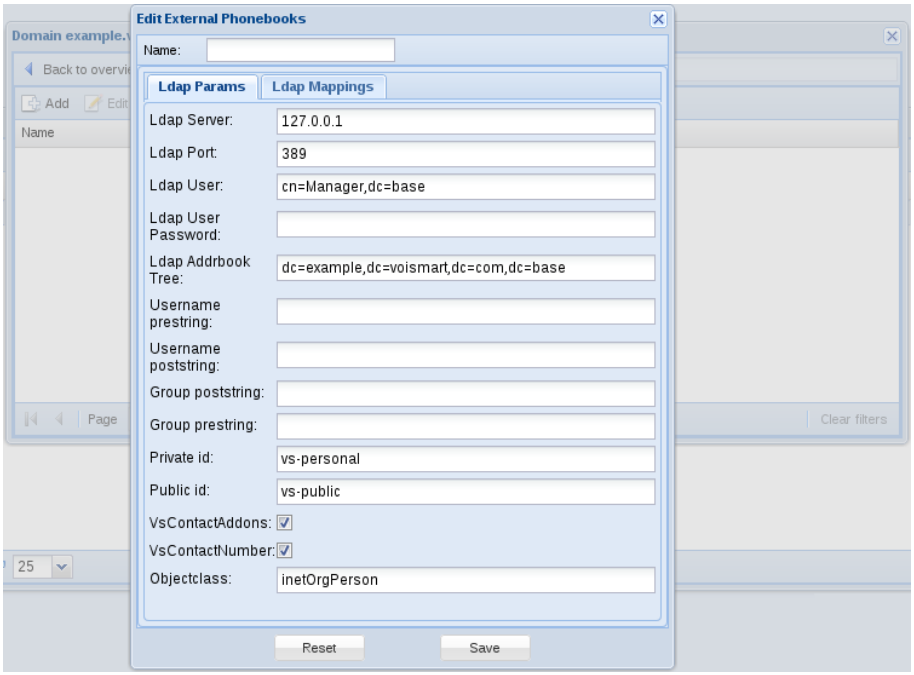

Figure 6.4: LDAP phonebook initial configuration tab

Ldap User Password password for the above username;

Ldap Addressbook Tree base search dn;

Username prestring prefix to attach to the contact entry owner

Username poststring suffix to attach to the contact entry owner

Group prestring prefix to attach to the contact entry group

Group poststring suffix to attach to the contact entry group

Private id value to assign to the owner attribute to mark the entry private

Public id value to assign to the owner attribute to mark the entry public

VsContactsAddons enable if the remote server has the VsContactsAddons additional schema;

VsContactsNumber enable if the remote server has the VsContactsNumber additional schema;

Objectclass objectclass of the directory entry.

Ldap Mappings

This section setups the mapping between server LDAP fields and internal Orchestra NG fields. On each field the key is the internal object attribute and the value is LDAP object attribute to read to fetch the actual data.

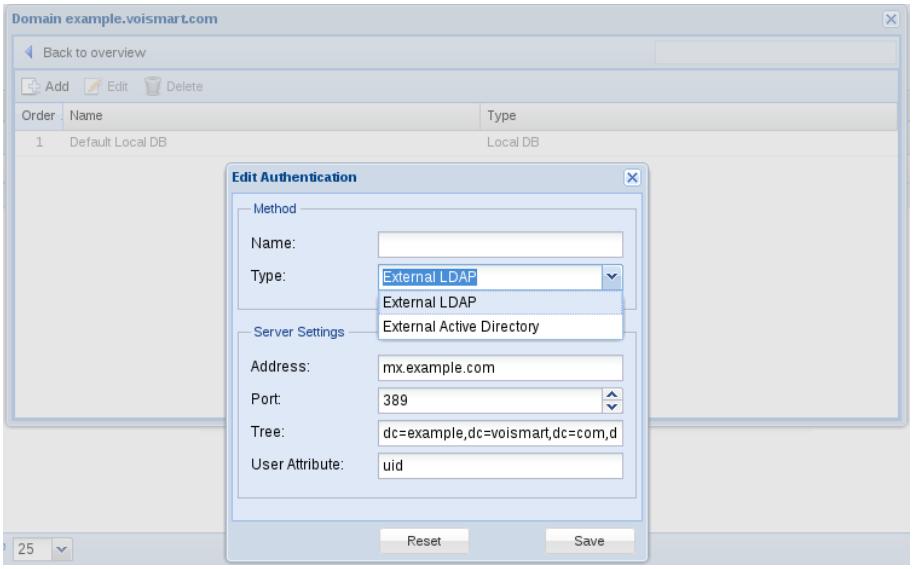

Figure 6.5: LDAP authentication configuration window

> ☞ Proper understanding of the LDAP protocol and remote server knownledge is needed. Mappings can vary a lot between different servers.

6.5 *External Authentication*

The external authentication feature allows to setup several external authenticators in addition to the internal one. All external authenticators must use the LDAP protocol. Support for Microsoft Active Directory is also provided. See figure 6.5 for a sample screenshot.

Name name or short description for the authenticator;

Type select between plain LDAP or Microsoft Active Directory;

Address hostname or IP address of the remote server;

Port remote server port;

Tree base search dn;

User Attribute attribute to check against for authentication.

6.6 *Autoprovisioning Groups*

This section is used to associate Autoprovisiong configurations, as defined in section 9.3 on page 76, to groups of supported phones belonging to the current domain. The group creation is shown in figure 6.6 on the next page and the supported phones are connected by selecting the action icon on the rule, as shown on figure 6.7 on the facing page.

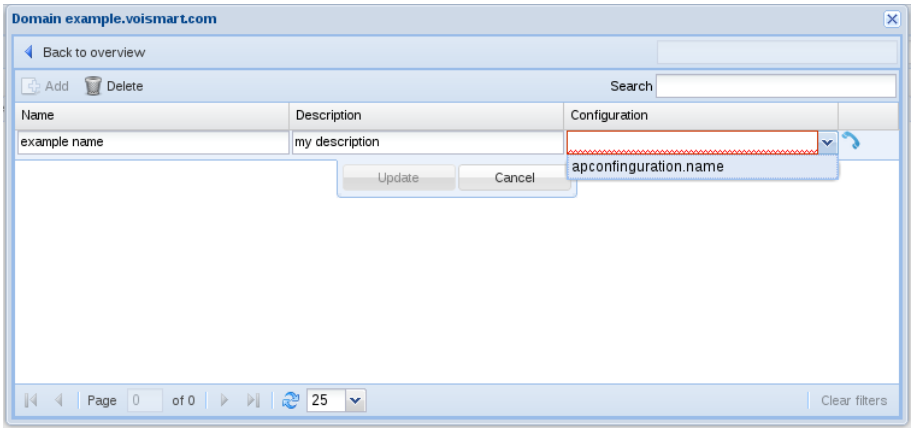

Figure 6.6: Domain autoprovisiong groups creation

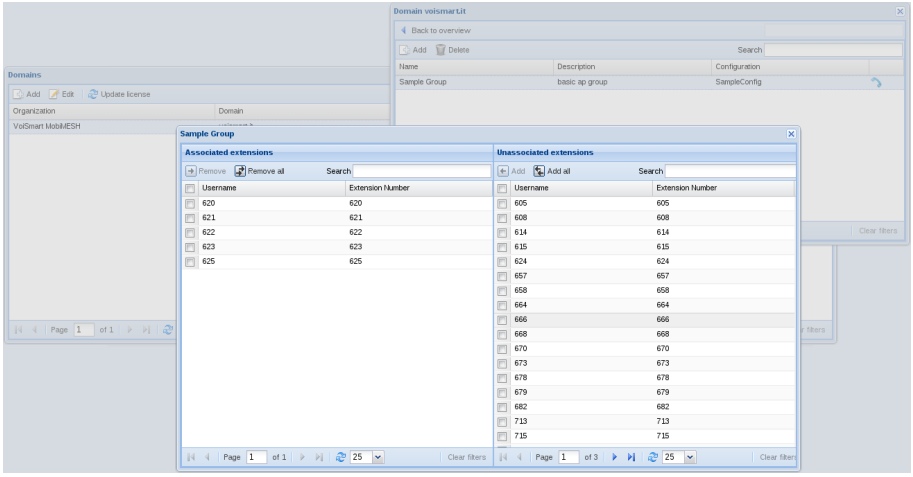

Figure 6.7: Association of autoprovisiong groups with extensions

6.7 *Network access point*

In this section is possibile to associate available system access points to the domain. Associating it means that is possible for user of the specific domain to logon on the Wi-Fi network using the same credentials of the Orchestra NG system, or setup a simple captive portal for the company; refer to user manual for captive portal setup. For configuring system access points, please refer to section 9.1 on page 74. For a sample association, see figure 6.8 on the following page.

6.8 *API Access*

The section allows to configure a list of trusted networks that can access a system API that allows to obtain a token on behalf of a specific user for that domain

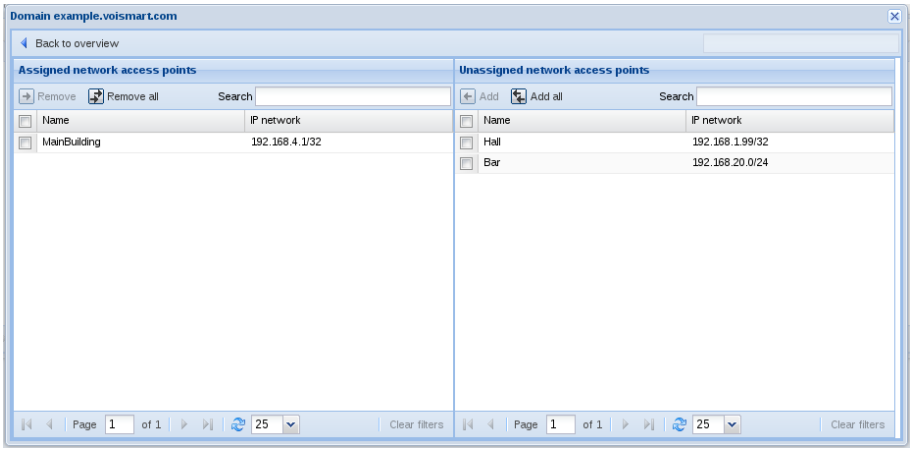

Figure 6.8: Association between system Access Points and domain.

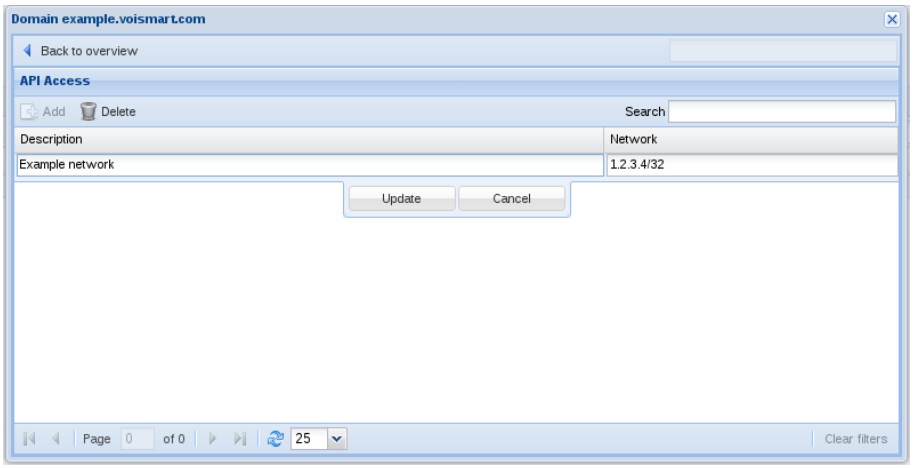

Figure 6.9: Domain API access configuration

without specifying the user password. The token permits to call subsequent APIs of the Orchestra NG system and perform several actions. This can be seen as a machine-to-machine trusted authentication. See figure 6.9 for a possible configuration.

7

Trunks

Contents

7.1 Concepts **44**
7.2 TDM interfaces **44**
TDM interfaces roles 44
Autodiscovery 44
Analog Interfaces 45
Basic Rate Interfaces 45
Primary Rate Interfaces 46
7.3 SIP profiles **48**
Best practices 48
Interaction with domains 48
Parameters 49
7.4 SIP gateways **50**
To register or not to register? 53
Parameters 53
7.5 NAT handling **54**

7.1 Concepts

In Orchestra NG trunks are general connections between the local system and a telecom provider, where the telecom provider can be a real telco or another SIP or TDM based system, for example another pbx.

7.2 TDM interfaces

TDM interfaces are physical PCI cards that allows to connect an Orchestra NG system to the PSTN network, using analog or digital technologies. A single interface can have multiple spans and be modular, where each module provides one or more spans.

A single span carries one voice channel for FXO or FXS interfaces, two voice channels for BRI interfaces and up to thirty for PRI interfaces.

Currently supported signalling protocols:

- FXS
- FXO
- Euro-ISDN

TDM interfaces roles

Each interface span can have three different roles:

master the span act as the network side of the link;

slave the span act as the customer side of the link;

reserved the span is not used and not configured.

> ✍ Unused or not connected spans must be configured as *reserved*.

Autodiscovery

The autodiscovery functions allows to detect installed TDM interfaces and setup them with basic parameters. See figure 7.1 on the facing page for a sample output of the discovery action, triggered by selecting the *Discovery* button.

In the above example, the system discovered:

- a dual port PRI interface (records 1, 2)
- a four port BRI interface (records from 3 to 6)
- a eight port hybrid FXO (records 7, 8) and FXS (record 12) interface. The reserved role means that neither FXO nor FXS port are enable (module missing).

Autodiscovery is able to detect also the *role* of analog and BRI interfaces, because is hardware based. Role of PRI interfaces is only software defined.

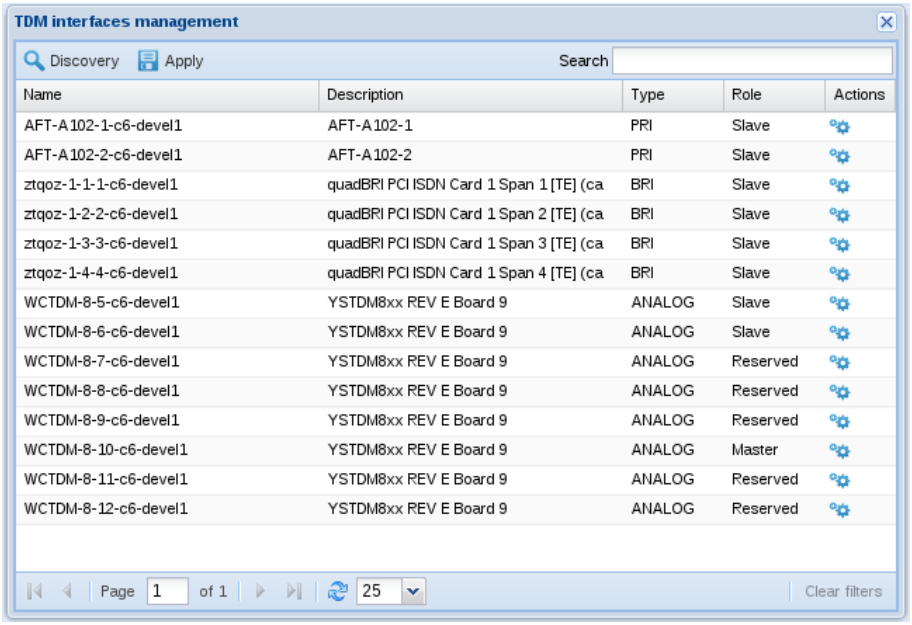

Figure 7.1: Example output of TDM interfaces auto discovery

Analog Interfaces

Analog interfaces are signalled by FXO or FXS protocol and each span provide one voice channel. Supported analog interfaces are modular, support up to 8 spans and can be hybrid, with mixed FXO or FXS ports. A FXO interfaces has a *slave* role, an FXS has the *master* role.

By selecting the *gears* icon under the *Actions* column, is possible to edit advanced span parameters.
FXS interfaces has no additional parameters.
FXO advanced parameters, see figure 7.2 on the next page:

DNIS is the DID to hunt on the system when a call come from this port;

Tone Group specifies the tone zone.

> ☞ *DNIS* must be always specified for FXO interface, otherwise incoming calls will not be routed.

Basic Rate Interfaces

Basic Rate Interface, or BRI for short, is a digital interface where each span provides two voice channels. This type of interface is signalled with Euro-ISDN protocol.

An interface with *master* role acts as the network side of the link. Network mode is also called NT mode. If *slave* role is selected the interface acts as the customer side. Customer side is also called TE mode.

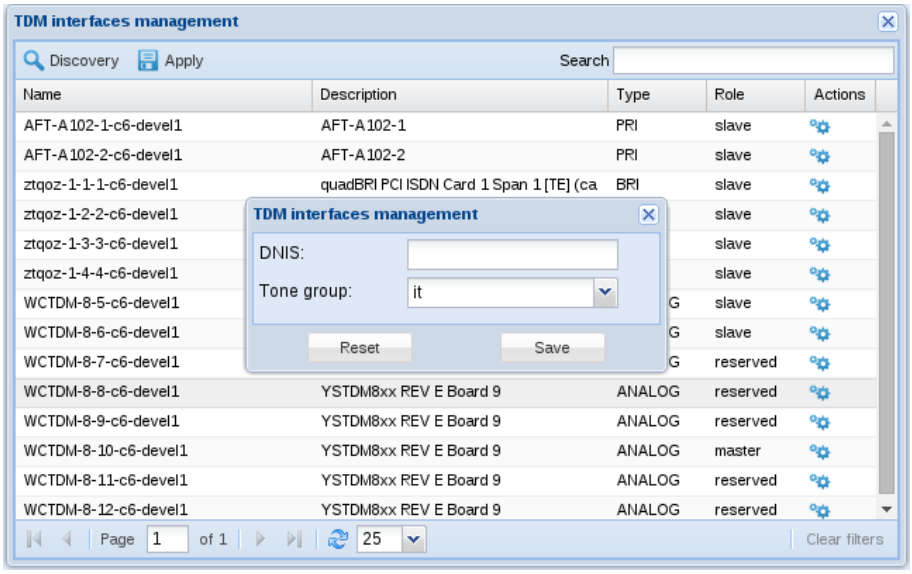

Figure 7.2: Analog interface advanced parameters

> ☞ To enable *master* role, jumpers on the PCI card must be set in NT position. Please refer to card manual for further details.

BRI interface advanced parameters, see figure 7.3 on the facing page:

Point2Multipoint if enabled sets up the link to Point-to-MultiPoint. Exact configuration depends on the local exchange;

Timer T302 is the sending complete wait timer. Refer to DSS1 specifications for detailed information;

National prefix appends the specified prefix to the caller id number on inbound calls, if the telco specifies the caller TON as national;

International prefix appends the specified prefix to the caller id number on inbound calls, if the telco specifies the caller TON as international.

Primary Rate Interfaces

PRI interface advanced parameters, see figure 7.4 on page 48:

Channel groups permit to create several virtual trunks using separate groups of channels. The channel groups are separated with comma and a channels range with the minus sign. Refer to section 8.4 on page 58 for usage of the trunks and virtual trunks.

> ✌ Indicating 1-15,17-20,21-30,31 will create four virtual trunks, the first one using channels from 1 to 15, the second one from 17 to 20, the third from 21 to 30 and the last one with only the channel 31. This is useful to reserve lines for inbound calls

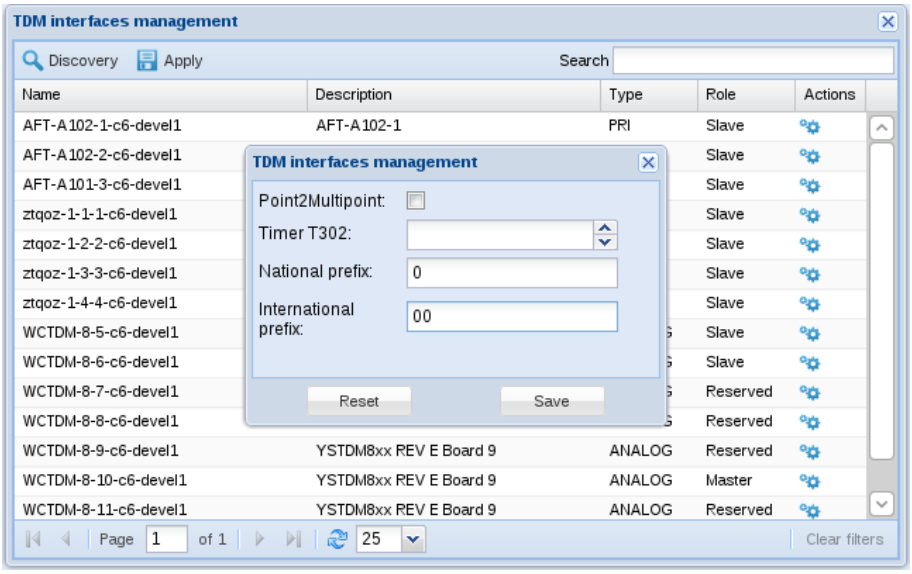

Figure 7.3: BRI interface advanced parameters

(by limiting the number of outbound channels) or assign a fixed number of channels to specific domains via Carriers/Lcr

Min digits Minimum number of digits to wait before considering the call complete. In overlap receiving scenarios, the call will be completed when a sending complete is received or minimum number of digits matched;

Timer T302 Sending complete wait timer. Refer to DSS1 specifications for detailed information;

Ref clock used to indicate the span number to use as sync source.

✍ Normally the *Ref clock* is not set if the role of the port is set to slave. This means that sync source is obtained from the NT side of the link, when the link is connected. A common scenario is to use a dual port card in pass-through mode, where the slave side is connected to the telco and has no *Ref clock* set, and the master side is connected to a downstream device. In this case the *Ref clock* for the master port is set to the span number of the corresponding slave port, resulting in having all the TDM path synchronized, from telco to downstream device. *Ref clock* can be set only between ports of same card.

☞ Not having a pass-through TDM path in sync, can lead to poor fax reception and voice glitches.

National prefix appends the specified prefix to the caller id number on inbound calls, if the telco specifies the caller TON as national;

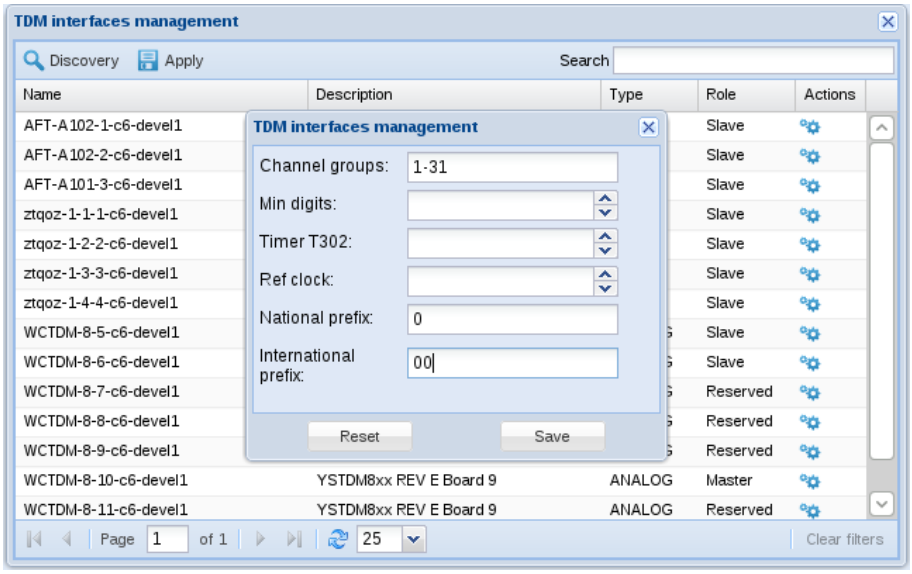

Figure 7.4: PRI interface advanced parameters

International prefix appends the specified prefix to the caller id number on inbound calls, if the telco specifies the caller TON as international.

7.3 SIP profiles

SIP profiles are instances of the Orchestra NG system, each one running with its own set of parameters. Running multiple profiles, or SIP stacks, is preferred against having a single one because it is easier to accommodate different scenarios which needs different parameters, make it possible to run different setups on different ethernet interfaces for multi-homed systems and so on.

Best practices

As a rule of thumb, one profile for each different SIP configuration is needed. Allocating different profiles with identical configuration is a waste of resources.

> One common setup is to have two different *SIP profiles*, one called *internal*, running on port 5060, used to interface with local extensions and a second one, let's call it *external*, running on port 5080, used to run any SIP trunk connected to an upstream provider.

Interaction with domains

Each *SIP profile* is a domain-less entity. It's just another SIP stack running. When a remote endpoint tries to contact Orchestra NG on a specific SIP port, the profile responsible for that port checks if the call needs to be authenticated. If it does, it gets the domain from the relevant URI, and looks up the calling

user on the users list of that domain, answering with a SIP authentication request. The same applies for the SIP registration process.

If a FQDN is used on the domain, it must be resolvable from the DNS system in use, otherwise extensions cannot reach the Orchestra NG system. Another approach when using a FQDN is to instruct phones to use the IP address of the pbx as outbound proxy: this allows to use the fully qualified SIP domains without having a proper DNS setup.

✌ The SIP domain is voismart.it. If some device tries to contact that domain, it will lookup the A record of the domain voismart.it using it's configured DNS. If this record exists, the phone will try to contact the address sent with the A record. In this scenario the A record must contain the IP address of the Orchestra NG system, so a properly configured DNS must exists.

✌ The SIP domain is voismart.it. If some device tries to contact that domain, it will lookup the A record of the domain voismart.it using it's configured DNS. The A record does not contain the IP address of the Orchestra NG system, or the domain voismart.it does not exist. The phone won't be able to contact the pbx, unless the phone has its Outbound Proxy configured to a valid IP address or hostname of the Orchestra NG system. In this scenario, the remote device will send SIP messages to the configured Outbound Proxy but will use voismart.it as domain for SIP requests and authentication.

Parameters

See figure 7.5 on page 51 for a basic sip profile configuration and figure 7.6 on page 52 for all the options.

***Profile name*:** name of the profile, any label;

***Profile description*:** short description for reference;

***Node*:** not used yet;

***Sip Port*:** port to listen to; the system listens to both TCP and UDP;

***Interface or IP*:** bind IP address or interface name. If IP or interface is invalid, the profile will not be started. Interface names depends on the system, normally they are eth0 for first ethernet, eth1 for second one and so on;

***External IP*:** if the system is behind NAT and connection to remote (i.e. not on the local LAN) devices is needed, the public IP address of the NAT'ing device;

***Auth Calls*:** enable authentication for incoming calls and allow authenticated register requests. Authenticated calls from an auth enabled profile will be treated as coming from local extensions. This flag must be active for *SIP profiles* used to connect with *local* SIP devices (phones, ATAs. . .);

***Force NAT*:** enable aggressive NAT detection. If *External IP* is set, this flag is automatically enabled internally;

***Enable 100rel*:** WARNING: Experimental feature! enable RFC 3262 support;

***Session Timers*:** enable support for RFC 4028 session timers;

***Point2Point RTP*:** WARNING: Experimental feature! point to point RTP. Allows to stay off the media stream, reducing bandwidth usage on the Orchestra NG system;

> ☞ Some features like transcoding between different codecs are lost and some services may behave incorrectly. Proper system and network knowledge, along with the proper features planning is needed before using this option.

***RTP Timeout*:** hangup the call if no RTP is seen for the configured interval. Value in seconds;

***RTP Hold Timeout*:** same as above, but calculated when a call is on hold. Since hold without media can last longer, it must be higher that *RTP Timeout*;

***TLS Port*:** specify which port is used for SIP TLS;

***Enable TLS*:** enable SIP TLS suppport;

***WSS Port*:** specify which port is used for SIP WSS;

***Enable WSS*:** enable WSS support for SIP, used with WebRTC clients;

***Cac Enabled*:** enable CAC feature on this profile. See caption 10 on page 81 for more informations;

***Cluster Communications*:** not used yet;

***Media Parameters*:** here codecs and their priority are defined, along with their applicable parameters. A *SIP profile* needs at least one codec. The priority is handled by ordering each entry by dragging it and dropping on the desired position. The G.729 codec needs an additional license, so be sure to have it before enabling, otherwise it will work only as pass-trough codec. Video codecs H.26X are only pass-trough codecs.

7.4 SIP gateways

A *SIP gateway* is a connection to another SIP endpoint, usually an upstream VoIP provider or gateway or another SIP device. A *SIP gateway* can be used in the LCR system to route outbound calls to it.

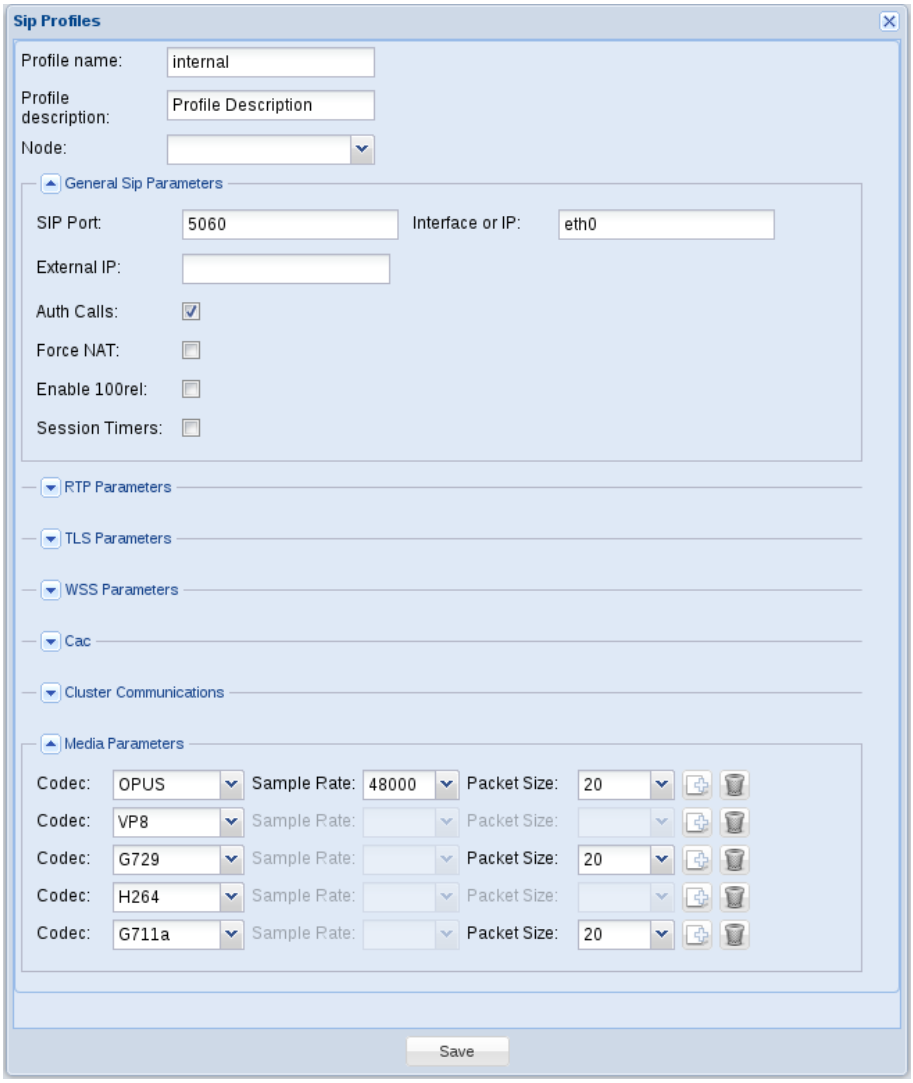

Figure 7.5: SIP profile basic setup

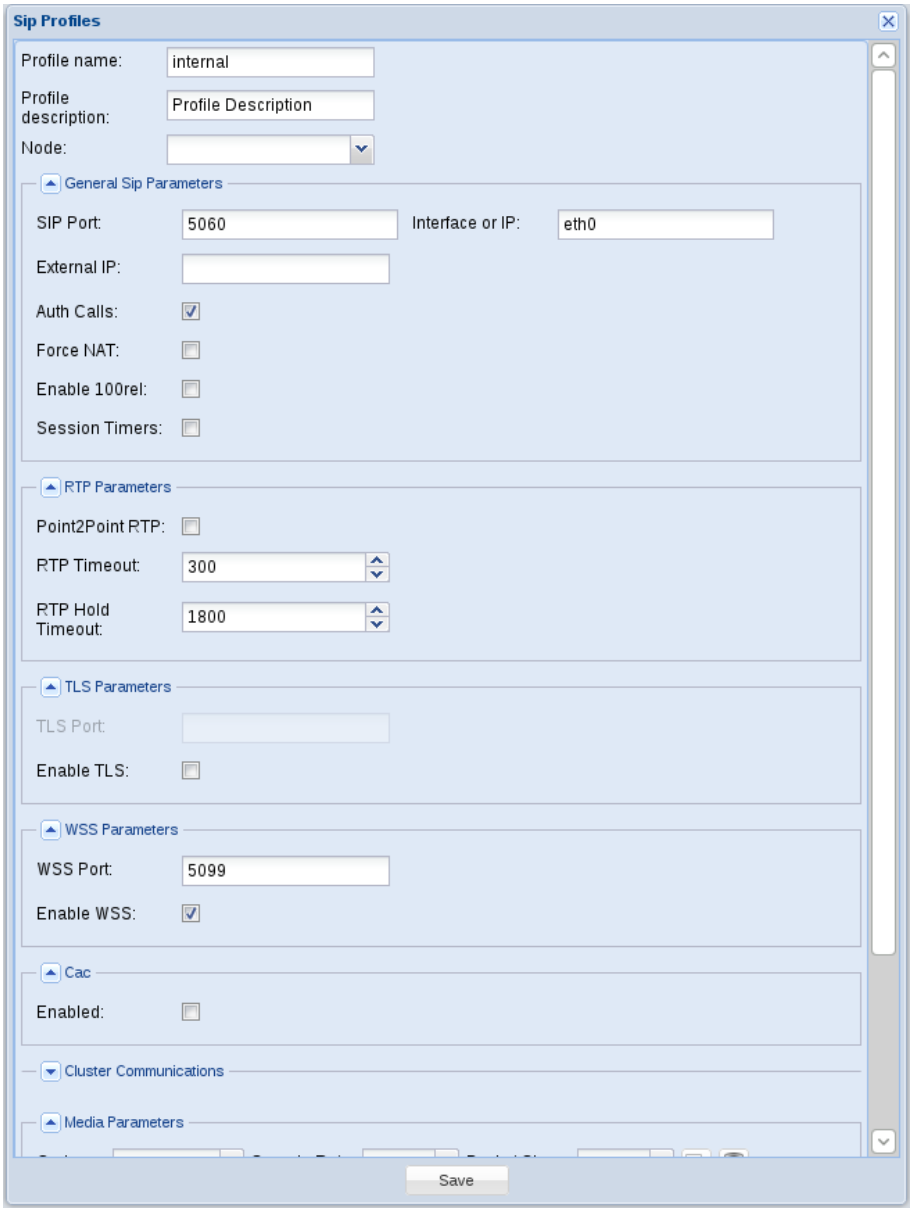

Figure 7.6: SIP profile advanced setup

To register or not to register?

Registering is a way to let the other endpoint know our IP address in order to reach us. A *SIP gateway* can be configured to register to the remote endpoint. This allows the remote endpoint to know where our system is and route calls to us. This is commonly used with VoIP services, because they do not know how to reach us. When the network scenario is known and all endpoints know each other, the registration is not needed. This is usually called as IP-IP SIP trunk, or SIP trunk for short.
19

Parameters

See figure 7.7 on page 55 for a *SIP gateway* sample configuration window.

***Name*:** a short name for the gateway;

***Description*:** optional description;

***Profile*:** on which *SIP profile* the gateway must be activated. This association is needed because it allows to specify different SIP parameters (the profiles) for different *SIP gateways*. It also affects the source SIP port from which outbound calls originate and the port the remote gateway will use if registration is enabled;

***Node*:** not yet used;

***Realm*:** SIP authentication realm. Used also as server name to contact if no *Proxy* or *Registrar* is specified;

***Proxy*:** SIP outbound proxy to route calls to. Port can be specified separating it from hostname with semicolon: e.g. my.proxy.com:5061; if port is not specified defaults to 5060; if not specified defaults to *Realm*;

***Registrar*:** if registration is enabled, a different registrar can be specified. Same rules as *Proxy* apply for SIP port. If not specified and registration is enabled, fallbacks to *Proxy* if specified or *Realm*;

***Auth username*:** authentication username for registrations or calls;

Auth password : authentication password for registrations or calls;

***Extension*:** extension used on registration which will be called with the upstream provider; if not specified defaults to the *Auth username*;

***Caller ID*:** put the callerid in the From header field on outbound calls via this gateway; most upstream providers seems to need this;

***Enable registration*:** enable registration to this endpoint;

***Register refresh*:** sets the registration refresh timer;

***Enable TCP*:** use TCP as transport protocol for this gateway;

***Trusted*:** whether to send P-Asserted-Identity or not. Enabling this flag will result into adding to outgoing calls a P-Asserted-Identity header with real user identity and Privacy header to declare the user requested privacy. If not enabled and privacy is requested, only the anonymous From header is set;

***Enable SIP ping*:** check the *SIP gateway* reachability using SIP OPTIONS method. If the ping fails or a SIP error message is returned, the gateway will be marked down;

***Ping frequency*:** how often to probe for the gateway status;

***Max forwards*:** if set, overrides the Max-Forwards value sent by the provider for inbound calls via this gateway; check section below for further details.

✍ Max-Forwards header is used to limit the number of proxies or gateways that can forward the request. Orchestra NG uses this value to prevent internal loops caused by misconfigured systems. Each hop inside the system (for example dialplan blocks) decrements the value by 1. When the value reaches 0, the call is dropped with a ROUTING_ERROR. Normally there's no need to set it, and if not set the value sent by the provider is used as starting point, since usually is high enough to accomodate all scenarios. With some provider the value is too low, so the call might be dropped during dialplan processing: if the provider value is 10 and the dialplan has 11 steps, the call will not reach the last one. In such situations, is possible to override it by setting the *Max forwards* item to a reasonable value, like 70. Setting it higher will result in a longer loop detection, if any, and higher resources usage.

7.5 NAT handling

The SIP protocol is not designed to cope with NAT flawlessly, so during the evolution of the protocol various mechanisms has been implemented to workaround SIP limitation on this topic. First of all, there's no one true answer to getting NAT traversal working. The reason is that different SIP endpoints, NAT, firewall settings and implementations mean what works somewhere might not work elsewhere. That of course makes it harder to manage clients at multiple sites, roaming clients, etc.

☞ Always disable SIP ALG on your firewalls or routers. It may not work or introduce unknown modifications to the signalling which just makes NAT debugging worse. If your trace shows a specific SIP packet being sent and on the receiving side the packet has been modified, there's a SIP ALG in between.

When on a specific *SIP profile* the flag *Force NAT* is enabled or the *External IP* is set, Orchestra NG enables a set of algorithms to work around NAT issues. *Force NAT* must be enabled if the system is not behind NAT but speaks with devices behind NAT.

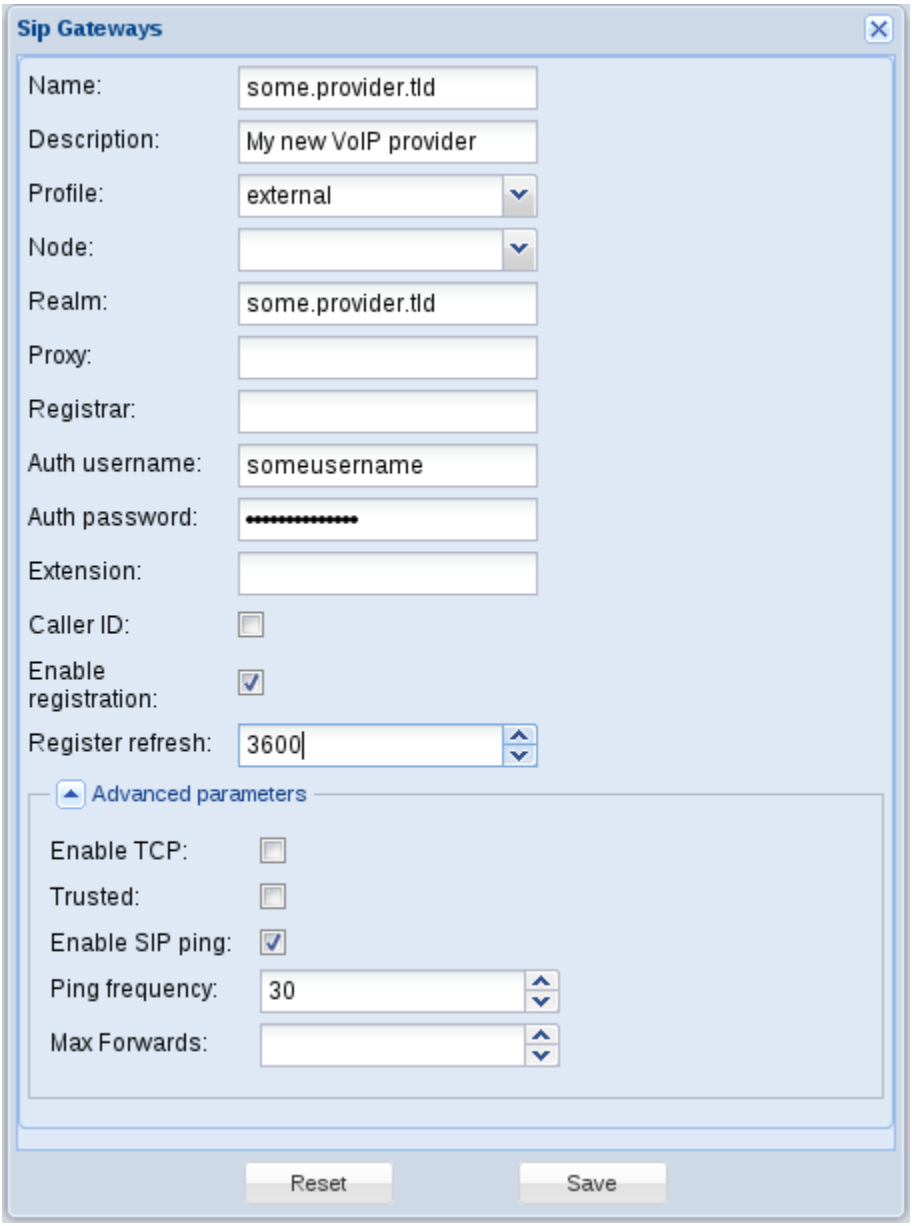

Figure 7.7: Example of *SIP gateway* basic configuration

✍ Some endpoints behind certain stateful firewalls may be able to call but not being called. Often it happens that the endpoint is callable and can be called just after SIP registration process, but when some time is elapsed it cannot be called anymore. This is because the stateful firewall is closing the NAT pinhole after an amount of time. The only solution here is to enable the ping option on the device. Refer to specific device manual on how to do that.

Orchestra NG is able to use UPnP to discover the external IP address. It allows auto NAT detection without any further configuration.

☞ If UPnP is not offered by the default gateway or multiple UPnP devices are present on the network, this can lead to NAT issues. Disable UPnP on all devices and proceed to a static NAT configuration using correct fields on the *SIP profile*. If unsure if the UPnP device works correctly, disable it.

8

Routing

Contents

8.1 Introduction 58
8.2 Concepts 58
8.3 *Inbound E164s* 58
8.4 *Carriers* 58
8.5 *LCR* 61
8.6 *Carrier maps* 65

8.1 Introduction

The *Routing* menu allows to configure inbound DIDs and their routing to different domains and LCRs for routing outbound calls.

8.2 Concepts

For inbound calls, for example calls coming from TDM trunks or unauthenticated SIP trunks, the system will look into *Inbound E164s* table to choose to which domain the call belongs to and route it to the specific domain's dialplan. Refer to user manual to check how to configure the tenant dialplan.

For outbound calls, a more complex routing is done. Each domain has one or more LCR associated to it, usable by the tenant users and when a call is placed, the correct trunk to use is selected using several decision steps. Refer to figure 8.1 on the next page to fully understand all the routing logic for outbound calls.

8.3 *Inbound E164s*

From the Routing → Inbound E.164s menu it is possible to open the *Inbound E164s* edit window like shown in figure 8.2 on page 60. On the left select the relevant domain and a list of configured rules will appear on the right. Select *Add* to create a new rule or *Delete* to remove an existing one.

***Rule Name*:** a brief, descriptive name of the rule;

***Rule Match*:** an expression to match the incoming number, using the regular expression syntax; see appendix C on page 97 for detailed information on how the regular expressions work;

***Define as fax*:** mark this number or rule as usable for fax server functions on the tenant. When activated a Station ID will be asked as default identity for faxes sent or received through this number/rule. The Station ID can be overridden on the tenant interface. See figure 8.3 on page 60 and Orchestra NG User manual for further details.

8.4 *Carriers*

From Routing → Carriers menu it is possible to associate one or more trunks to a carrier. In addition it is possible to disable it or define a monetary cost for every destination reached by the specific carrier. By selecting *Add* button is possible to create a new Carrier. Double clicking on an existing entry allows to edit it. See figure 8.4 on page 61 and 8.5 on page 62 for reference.

> ✍ Each time a sip trunk is created, a new carrier is automatically added, with the name beginning with Auto_, in disabled state and with the corresponding sip trunk already associated.

***Name*:** a descriptive name;

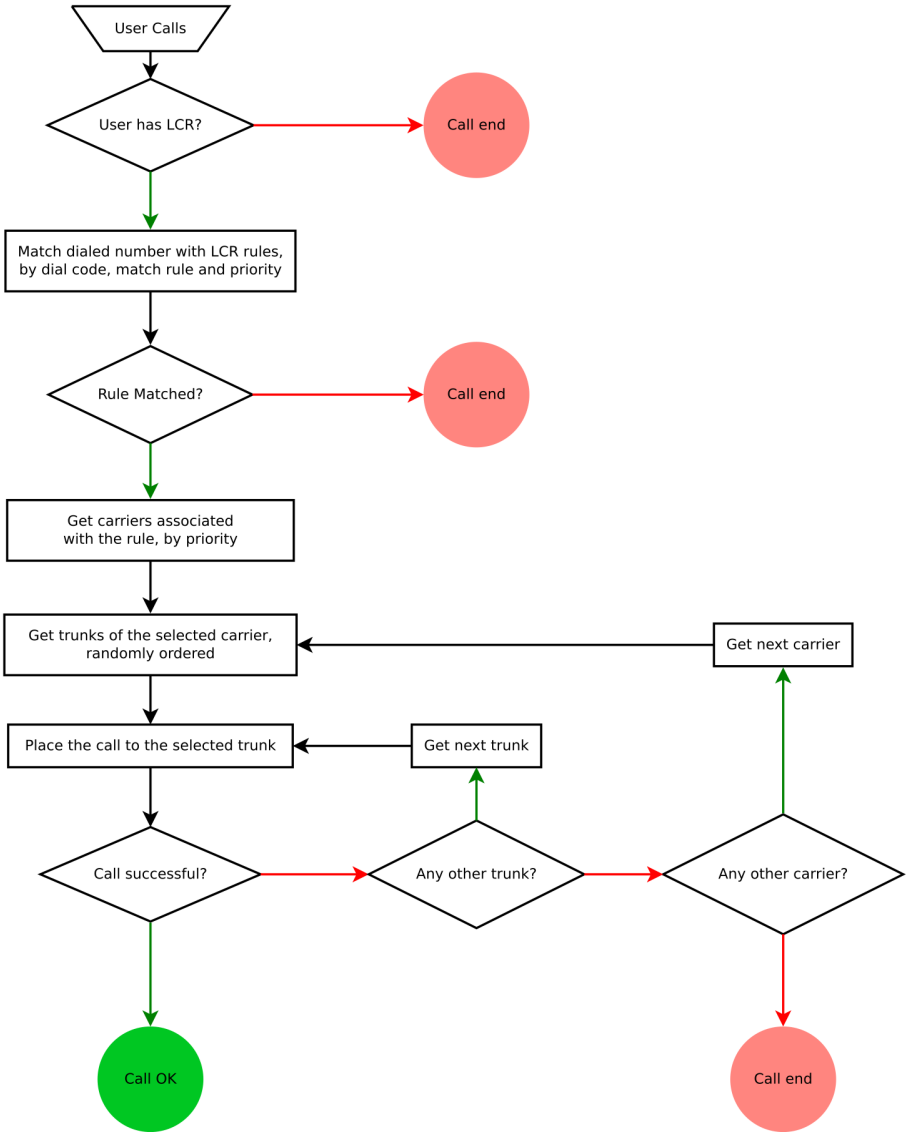

Figure 8.1: Schema of LCR routing logic

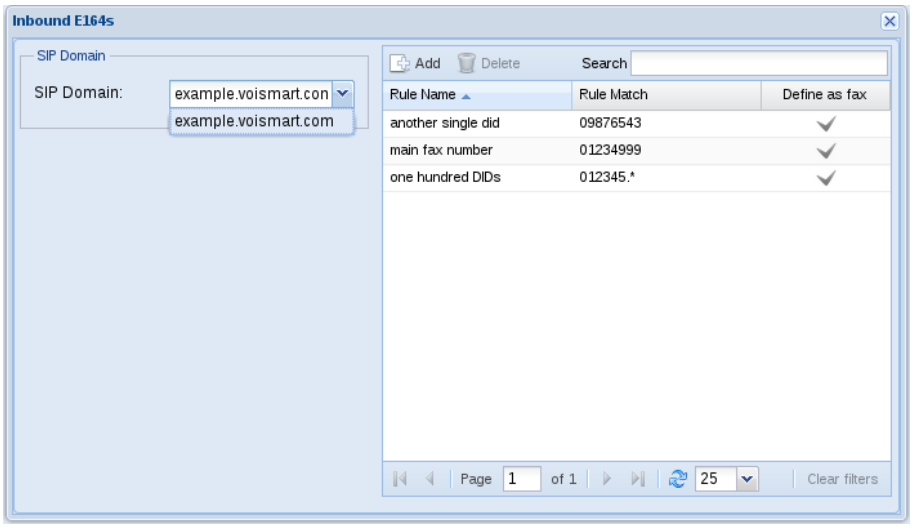

Figure 8.2: E164 numbers setup window

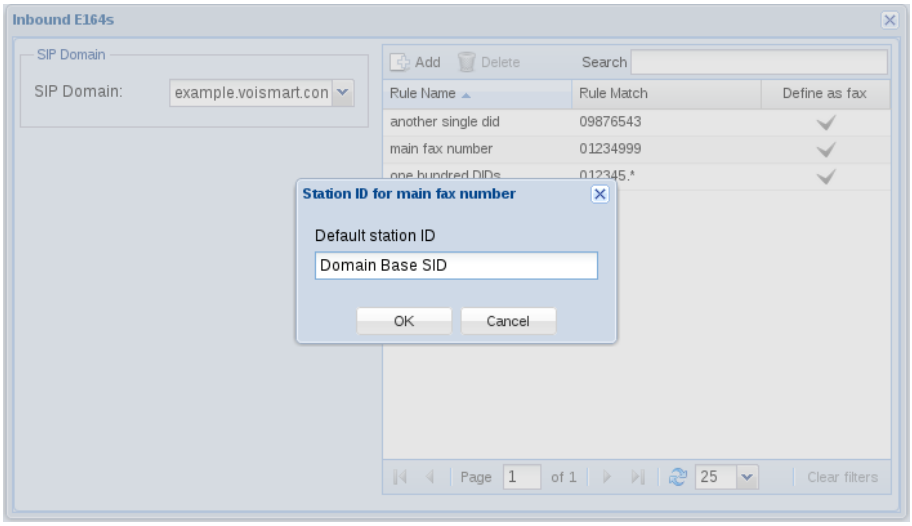

Figure 8.3: E164 fax station ID setup window

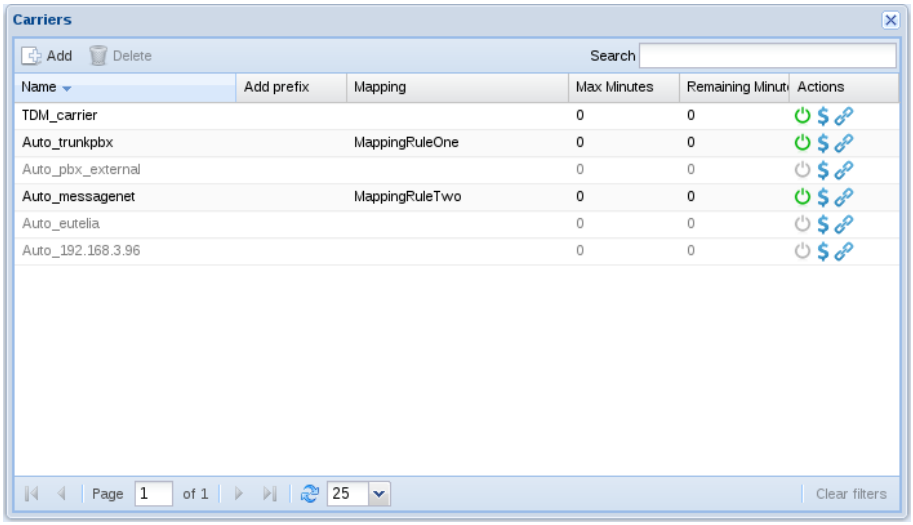

Figure 8.4: Carriers setup window

***Add prefix*:** prefix to add to all numbers dialed trough this entry;

***Mapping*:** mapper to use to manipulate outbound caller id, see 8.6 on page 65;

***Max Minutes*:** not yet implemented;

***Remaining Minutes*:** not yet implemented;

***Action Disable*:** disable this entry;

***Action Rates*:** not yet implemented;

***Action Associate*:** associate one or more trunks to this entry, see figure 8.5 on the following page.

It is possible to associate one or more trunk to each carrier. If several trunks are associated, when calling trough the carrier, a trunk is selected in a random fashion and the call placed. If the trunk fails (because is unreachable or congested), another one is used from the associated trunks.

> ✍ Associating multiple trunks to each carrier allows to easily build a load balancer and failover between trunks.

8.5 *LCR*

An LCR is used to route outbound calls to different carriers, or group of carriers, using the called number prefix for routing decision. It is possible, for example, to route mobile phone calls to a sip carrier and landline calls over TDM trunks. Is also possible to route each destination to multiple carriers and trying them in an orderable fashion. In addition, if each carrier has more than one trunk associated, each trunk is used until the call is successful.

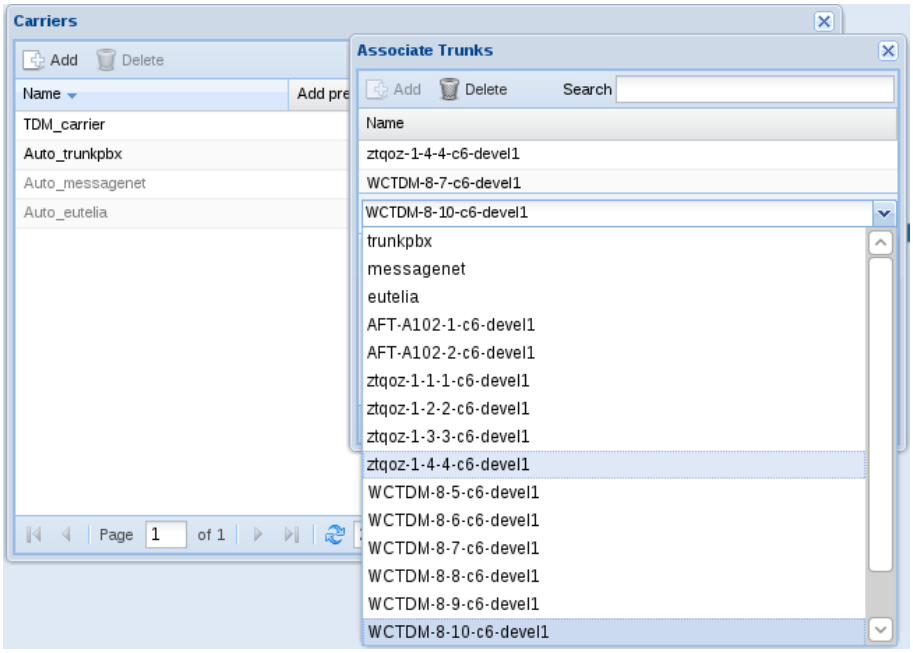

Figure 8.5: Carriers association with trunks

From Routing → LCR menu it is possible to manage existing LCRs and to create new ones. First step is to create a new one using the *Add* button, then associate it to one or more domains, in order to be usable by the specific associated tenants (for example a tenant can have more than one LCR and assign different LCRs to different users within the domain). See figure 8.6 on the next page and figure 8.7 on the facing page for reference.

After selecting the LCR and pressing the *Next* button it is possible to edit the LCR rules, as shown figure 8.8 on page 64. Each rule is a pattern match against the dialed number and, if it matches, the call will be routed to the associated carriers. The match priority can be adjusted by dragging the rule row in the desired position and dropping it.

> ✍ Each time a new LCR is created a default LCR rule is automatically added.

***Priority*:** the rule priority, can be modified by dragging the rule and dropping it in the desired position;

***Match Rule*:** a regexp against which the dialed number will be matched;

***Dial Code*:** a prefix to select the rule to strip before dialing; takes precedence over match rule;

***Day Start*:** from which day of the week the rule is valid;

***Day End*:** until which day of the week the rule is valid;

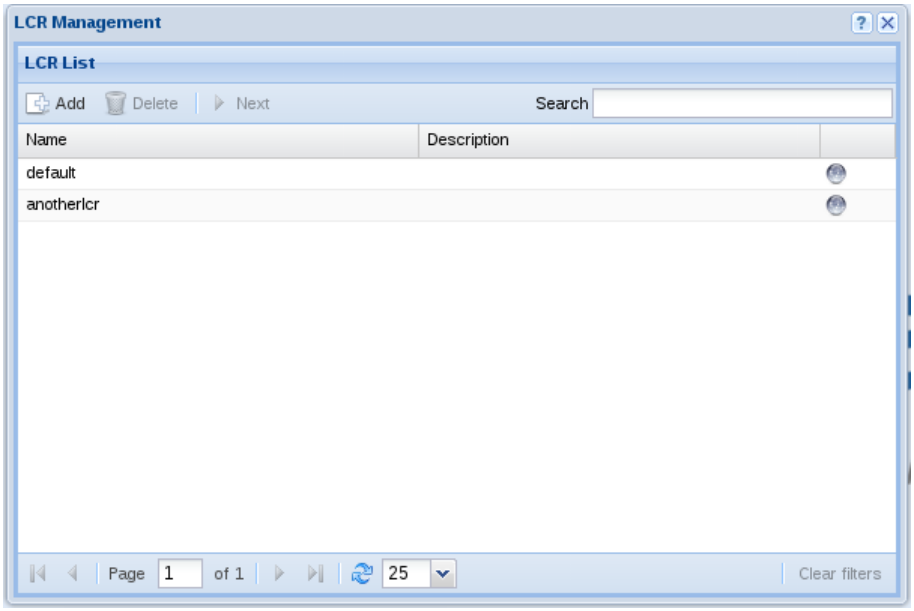

Figure 8.6: LCR setup window

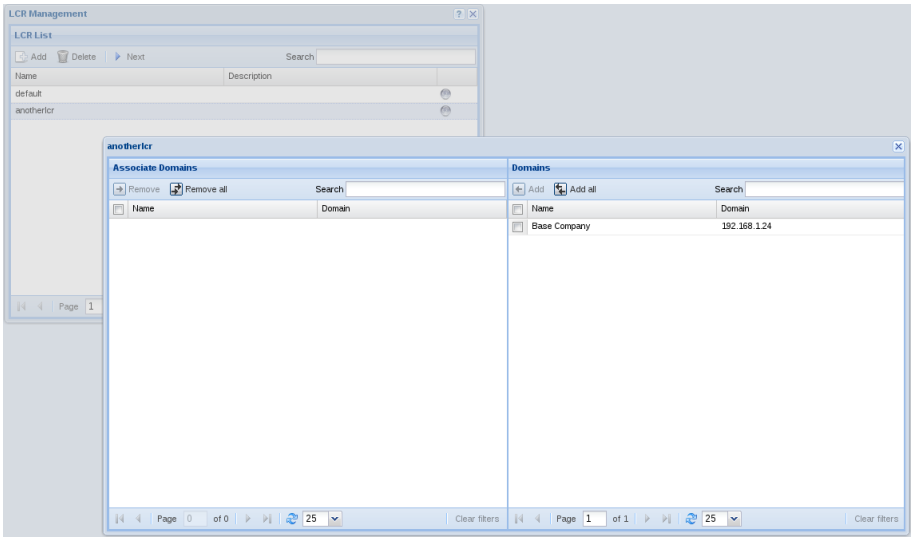

Figure 8.7: Association between LCR and tenants

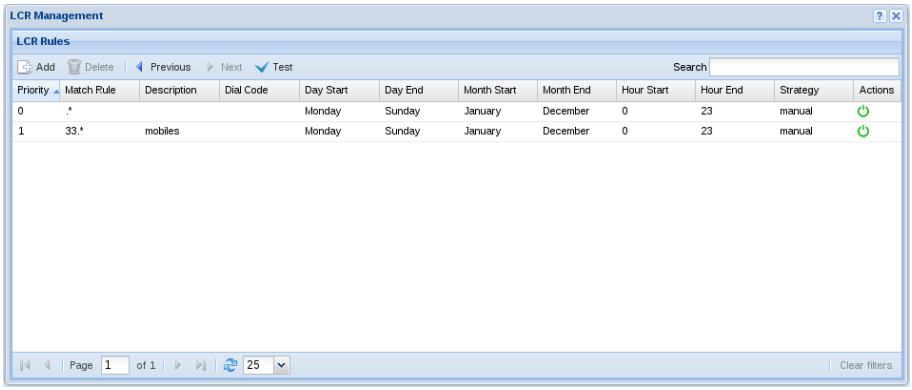

Figure 8.8: LCR rules configuration

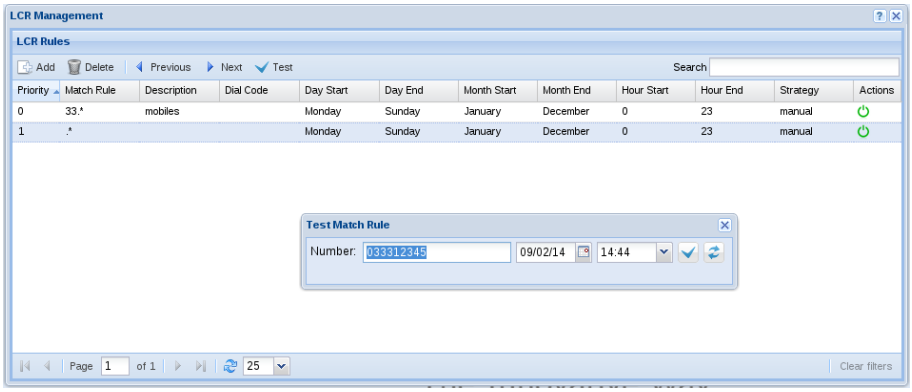

Figure 8.9: LCR rules test function

***Month Start*:** from which month the rule is valid;

***Month End*:** until which month the rule is valid;

***Hour Start*:** from which hour the rule is valid;

***Hour End*:** until which hour the rule is valid;

***Strategy*:** only manual strategy is supported right now;

***Action Enable*:** enable or disable the current rule.

By using the *Test* button it is possible to check which rule is engaged for a specific number. Just write the number, select the date and time and press test. The selected rule will be highlighted in the LCR rules window. See figure 8.9 for reference.

The Final step is to associate one or more carriers to each rule. To do this, select the appropriate rule and press the *Next* button, and the association panel will appear, as shown on figure 8.10 on the next page. At least one carrier is needed for the rule to work, otherwise the call will be dropped, since the rule has no exit point. If multiple carriers are added, they can be ordered by dragging

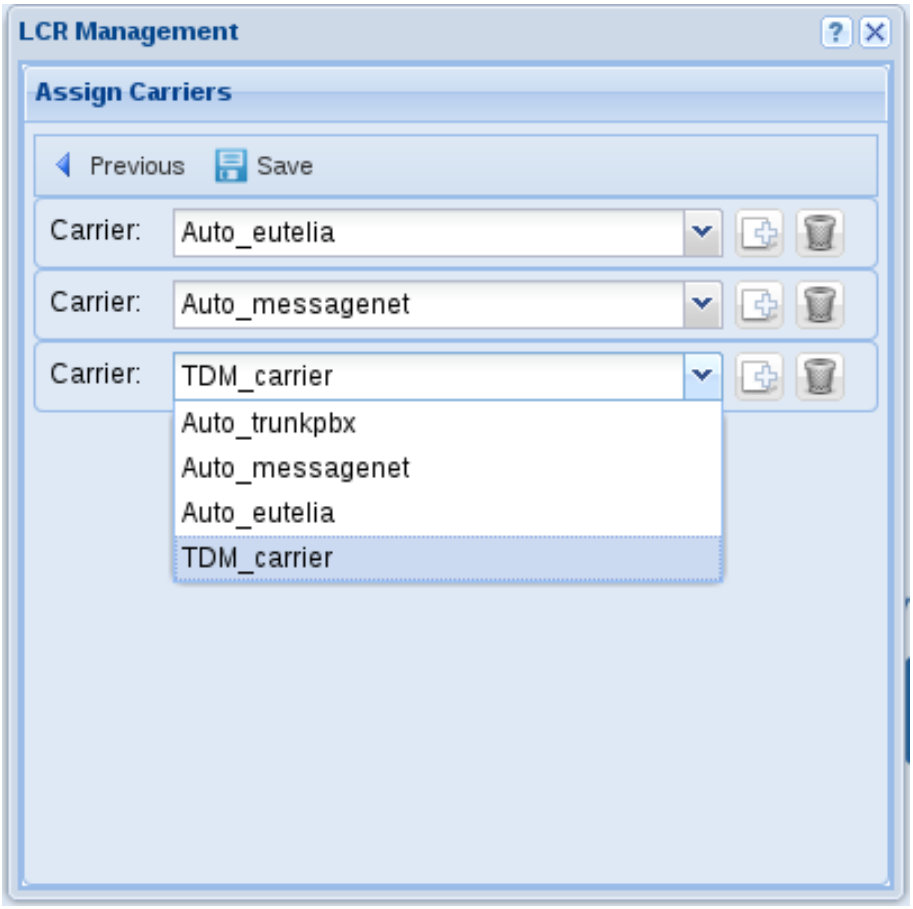

Figure 8.10: LCR rules association with carriers

and dropping in the correct position. When multiple carriers are selected, they will be used in the order shown, and the system will automatically route the call to the next carrier if the first is unreachable or congested. When the carriers are added or modified, press the *Save* button to save the association configuration.

8.6 *Carrier maps*

Carrier maps allow to remap caller numbers in a powerful way, using regexp substitution, see appendix C on page 97. It consists in a set of rules which can be applied to outbound calls, rewriting the caller number. This substitution happens immediately before placing a call on a specific carrier. If an outbound map has been defined in the user interface, the match will be done against the outbound map transformed number. Using carrier maps allows to setup different caller id number rewriting rules that depends on which carrier is used.

To create a new mapping, open the main configuration window (figure 8.11) by clicking on the Routing → Carrier maps.

In that dialog, click on the Add button and create a new mapping using

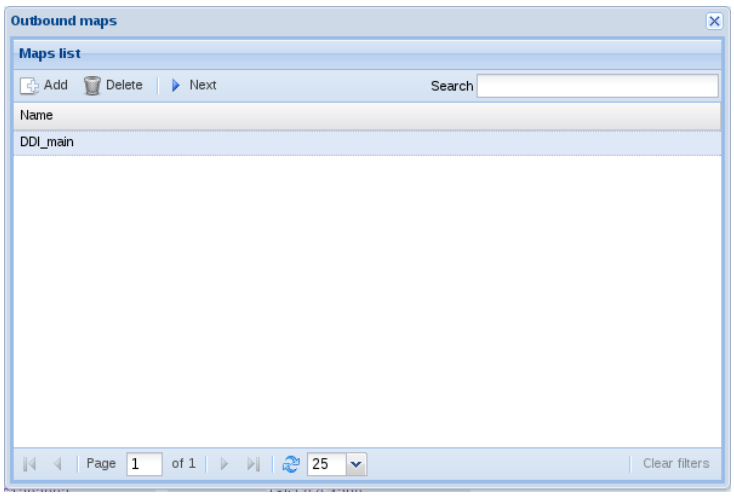

Figure 8.11: Main carrier maps window.

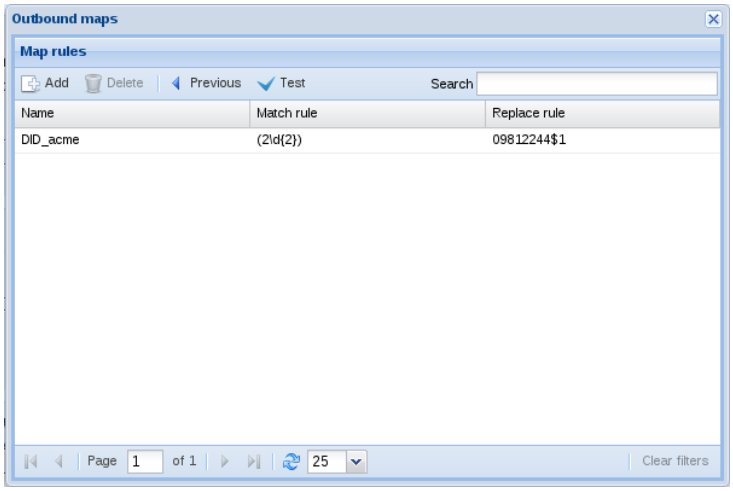

Figure 8.12: Carrier maps rules window.

a descriptive name. To add new rules, select it and click on the Next button to enter the outbound maps rules dialog, figure 8.12.

New rules can be created and destroyed using the Add and Delete buttons. A new rule consists in a descriptive name, and the match and replace patterns. The *Match rule* is the pattern against which the caller number is matched. If a match is found, the *Replace rule* pattern is used to rewrite the caller number, using the set of rules explained in appendix C on page 97.

Rules can be quick-tested by using the Test button and inserting the number to test and pressing *Ok*. The replaced number will be shown in a dialog, or a brief message will appear if no rules matches that number.

Carrier maps are only used by the carriers associated to them using the *Mapping* field in the carrier configuration dialog, see section 8.4 on page 58.

> ☞ Remind that caller numbers replaced by carrier map works only if it matches with inbound E.164 numbers defined in the domain which is currently placing a call. If there is no match, caller's number is marked as private.

To delete a carrier map, just select it and click on the Delete button 🗑.

9

System

Contents

9.1 System Settings **70**
Mailserver 70
Remote Support 71
Firewall 72
Cac 74
WiFi 74
9.2 Backup **75**
9.3 Autoprovisioning **76**
Autoprovisioning server 78

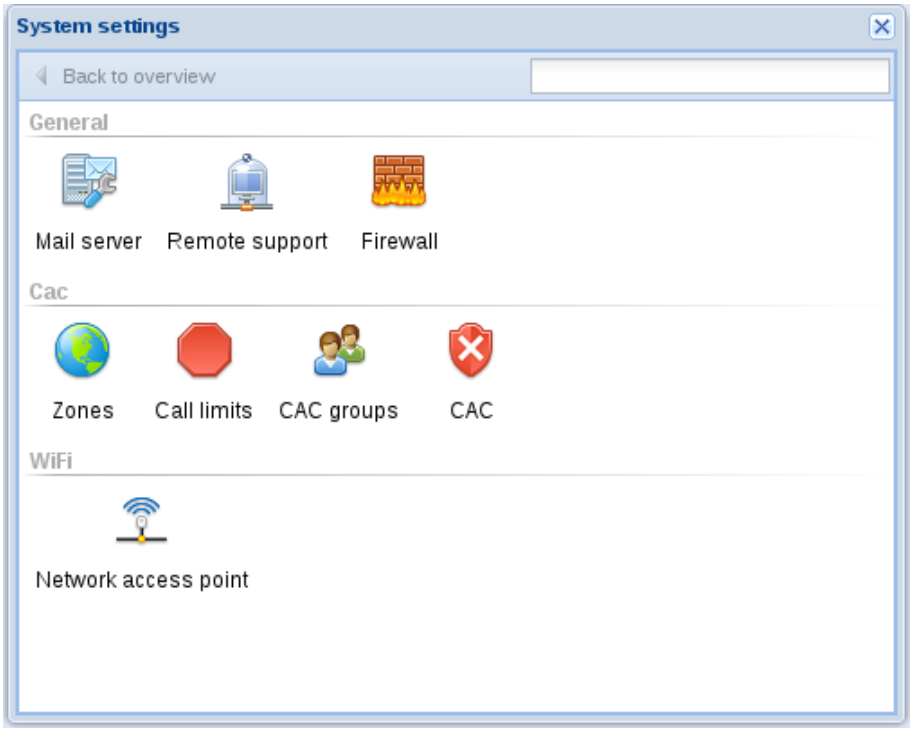

Figure 9.1: System settings panel

9.1 System Settings

Selecting the System → System Settings menu brings up the system settings panel, as shown in figure 9.1. From here it is possible to select the item to configure and proceed to the next panel.

Mailserver

The mail server configuration is needed for proper delivery of all notifications. Internally all services deliver emails to a local server which in turn needs a real, fully qualified mail server to use as an upstream smarthost. By selecting the Mailserver icon, a new panel appears:

***Hostname*:** hostname of the local mailserver, used in HELO commands;

***Sender email*:** default sender email, if not specified in the notifications configuration;

***SMTP server*:** name of the smarthost or the upstream SMTP server which will relay the Orchestra NG emails;

***SMTP port*:** port of the upstream smarthost, 25 by default;

***Encryption type*:** whether to encrypt the connection or not; TLS is strongly suggested for security and privacy;

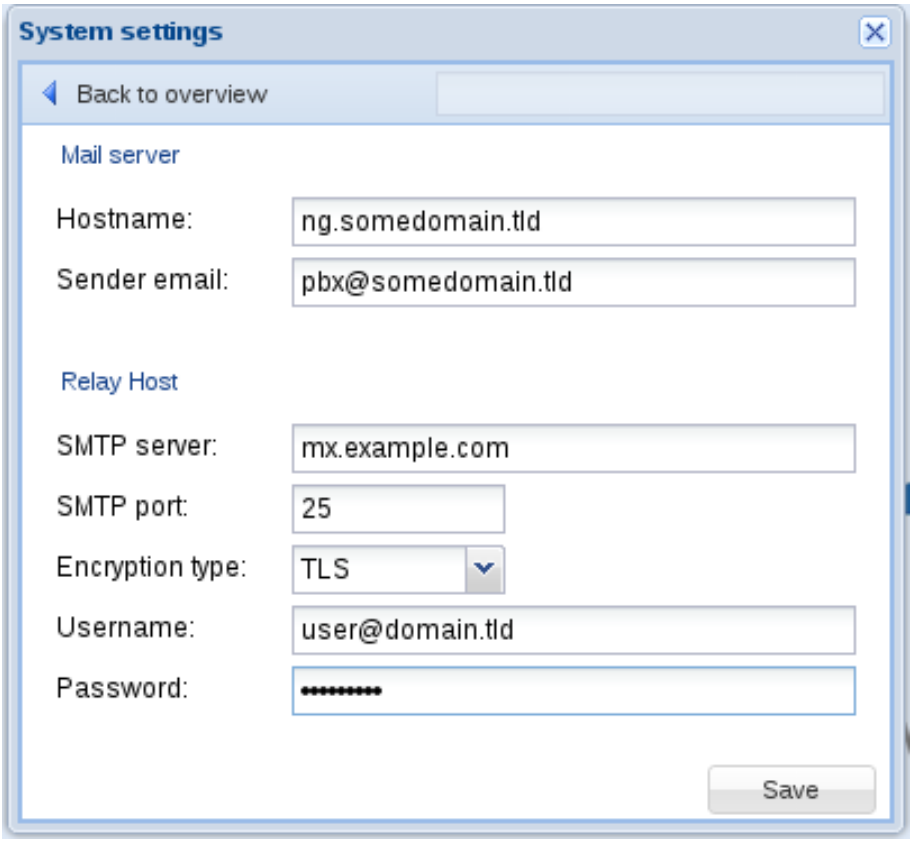

Figure 9.2: Mailserver settings panel

***Username*:** if the upstream server needs an autheticated submission, write here the username;

***Password*:** password of the above username, if needed.

Refer to figure 9.2 for an example configuration.

Remote Support

If provisioned with proper certificates, the Orchestra NG system can connect to a central VPN server operated by VoiSmart for support purposes. If connected to the central server, VoiSmart personnel can connect to the system more easily than opening firewall ports and setup port forwarding on the customer side. By selecting the *Enable* button, a connection attempt is done and if successful a connected state is shown, as long the IP address assigned to the VPN tunnel, like shown in figure 9.3 on the next page for an example.

> ✍ By default no certificate is installed, only when a support contract is signed with VoiSmart the certificates are provisioned by the helpdesk techs.

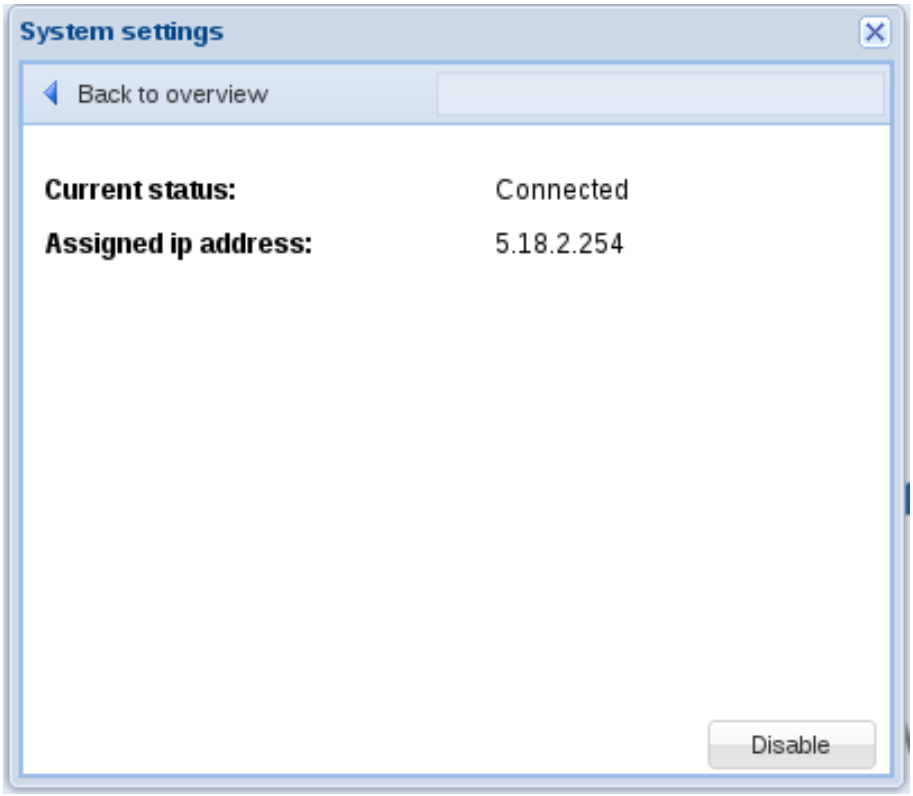

Figure 9.3: VPN panel

> Even with the tunnel enabled VoiSmart needs the proper usernames and passwords to access the command line terminal and/or the web gui, there's no other way for support services to gain access to the system.

Firewall

By selecting the Firewall icon, it is possible to configure the built-in firewall subsystem. The built-in firewall allows to setup proper rules to protect the system from unwanted accesses, restrict access to service only from predefined networks, rate limit inbound traffic and provides basic filtering for SIP registration traffic. See figure 9.4 on page 74 for an example configuration.

To enter the configuration, select the hostname from the drop-down list.

> ✍ If the hostname is changed, the firewall must be reconfigured!

From the panel, it is possible to edit, add or delete rules. Since firewall rules follows a priority from top to down, it is possible to reorder them by drag each row and drop it into the desired position.

***Source address*:** source network address to filter, in CIDR format. Use /32 to filter a single host, 0.0.0.0/0 for all traffic;

***Destination port*:** port or range of inbound ports to filter, in port[,port:port,port...] format; for example 5060, 16384 : 20000 will filter port 5060 and port range from 16384 to 20000;

***Dst interface*:** limit the match to the specified interface, by using the linux interface name; normally this is something like ethX where X is the number of the interface, like eth0;

***Protocol*:** which protocol to filter, can be TCP or UDP; if protocol is not specified, ports cannot be set;

***Policy*:** which policy to use for the match, see below for further details;

***Parameters*:** optional parameters for the policy, depends on each policy;

***Actions*:** enables or disables the rule without deleting it;

When a rule matches, a policy is executed. If not specified, the rules processing stops at first match. The following policies are supported:

***ACCEPT*:** accepts the traffic;

***DROP*:** discard the received packets;

***REJECT*:** like DROP, but sends back an ICMP port unreachable message; this is a more polite way of filtering traffic but can consume upload bandwidth if a lot of traffic is being rejected;

***RATE_LIMIT*:** rate limits the traffic and drop only if certain thresholds are reached. If the thresholds are not reached, the rule processing will go on;

***SIP_REG_CHECK*:** checks for SIP registration attempts and failures and drops traffic from the source if thresholds are met, for a configurable amount of time;

***SIP_REG_WHITEL*:** white lists the specified networks from the SIP_REG_CHECK, to be used before the SIP registration check, useful when receiving a legitimate amount of attempts from a known source.

SIP_REG_CHECK and *RATE_LIMIT* takes parameters, described below:

***RATE_LIMIT*:** the parameter takes the format S/H where S is the observation window and H is the hit count, or how many packets are seen from a single source in the observation window; if more than H packets are seen in the S window, the traffic is dropped until it stops, otherwise rule processing continues. S is in seconds, H is an integer value, with maximum value of 20. 5/20 means drop traffic if more than 20 packets are seen in 5 seconds from a single source;

***SIP_REG_CHECK*:** the parameter takes the format $R/S/B$, where R is max retries, S the observation window and B is the ban time; 5/60/300 means that if a single source attempts or fails to register for more than 5 times in 60 seconds, it will be banned for 300 seconds. Since the expiration task is not immediate, the ban time can be up to 60 seconds longer (360 in the above example).

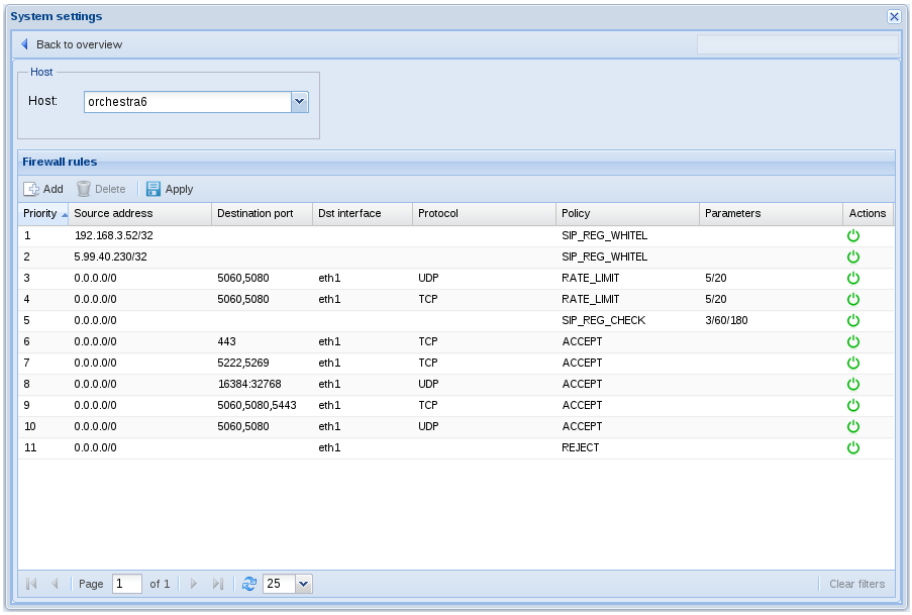

Figure 9.4: Firewall configuration panel

For the ports used by various services, refer to 11.4 on page 90.

When the firewall configuration is complete, click on the *Apply* icon to activate the configuration.

> ☞ Pay attention to not lock out your connection with the system, otherwise manual intervention and physical access to the machine will be required to disable the firewall!

> ✍ After reboot, during services startup, the firewall is disabled for a brief time, until the login form is shown on the web gui. This is meant to provide an emergency access window to the system to recover from wrong firewall rules.

Cac

Because CAC is a complex argument, please refer to the specific chapter 10.

WiFi

The Orchestra NG system can act as a radius server for Wi-Fi access points supporting it, in order to authenticate users using the same Orchestra NG credentials.

Supported authentication schemes are:

- PEAP-MSCHAPv2: supported by all devices and operating systems that support EAP;

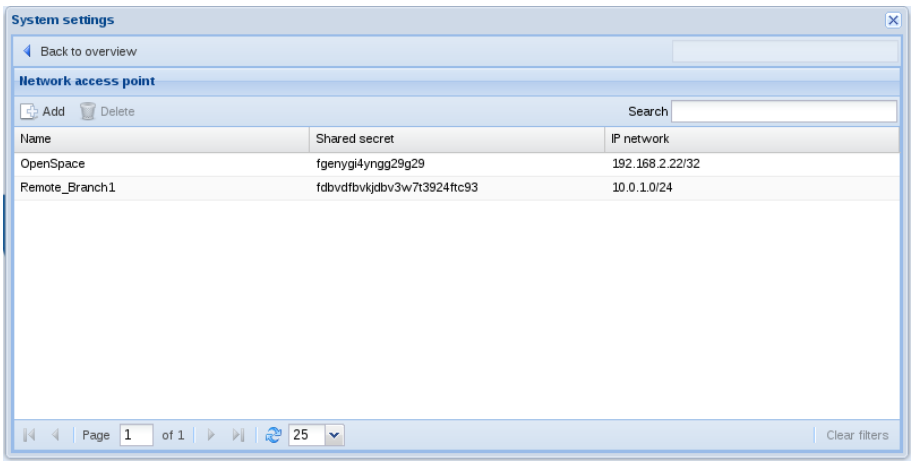

Figure 9.5: WiFi access points credentials setup

- EAP-TTLS-PAP: supported by the vast majority of devices and operating systems, notably excluding Microsoft Windows pre 8, Windows Phone pre 8.1 and iOS if not provisioned by a configuration server.

As shown on figure 9.5 only three parameters are needed to setup a radius access:

***Name*:** a descriptive name of the entry;

***Shared secret*:** the pre shared key used from the access point(s) to connect to the radius server;

***IP network*:** from which IP address or network radius clients can connect. If a network is specified, all clients within this network share the same secret.

After configuring one or more access points they can be associated to tenants. See section 6.7 for further details.

9.2 Backup

From the System → Backup it is possible to export and import the system configuration for backup purposes. See figure 9.6 on the next page and figure 9.7 on page 77 for reference.

By selecting the proper tab it is possible to switch between import and export functions.

To import a backup, select the *import* tab, use the *Browse* button to select a previously downloaded backup file and then select *Import*. Wait until a popup confirms the operation.

To export a backup, select the *export* tab and select *Export*. Wait until a popup confirms the operation and the file will be automatically downloaded.

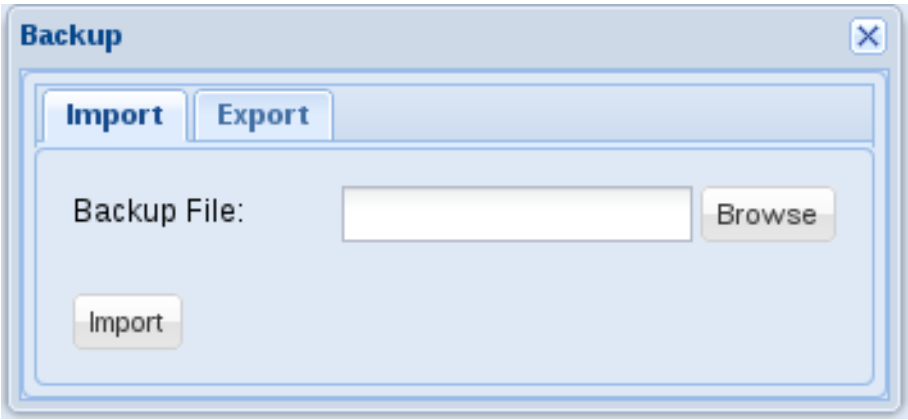

Figure 9.6: Backup import

✍ An import or export operation can take a long time, depending on backup size and underlying hardware.

✍ Only system configuration is exported, except network configuration and underlying Linux configuration. Basically only configurations managed from web gui is exported.

Data produced by the normal work of the Orchestra NG system is not exported, since it is not a configuration. Such data is sent/received faxes, voicemails, cdrs, fdrs, stats and so on.

☞ If the imported backup belongs to an older version of Orchestra NG, a reboot is required after the import operation.

Backups from newer versions cannot be imported.

Do not import backups during heavy use of the system, since some functions may be restarted or stopped automatically. For importing a backup is always a good idea to schedule a maintenance window to avoid service disruption.

Interrupting a backup import may lead to database corruption and the system must be restored manually by trained personnel.

9.3 Autoprovisioning

Orchestra NG can provision specific SIP phones, using the HTTP or HTTPS protocols only, and several configurations can be created, all phone-independent, for later association to phones or groups of phones in the tenants, as described in section 6.6.

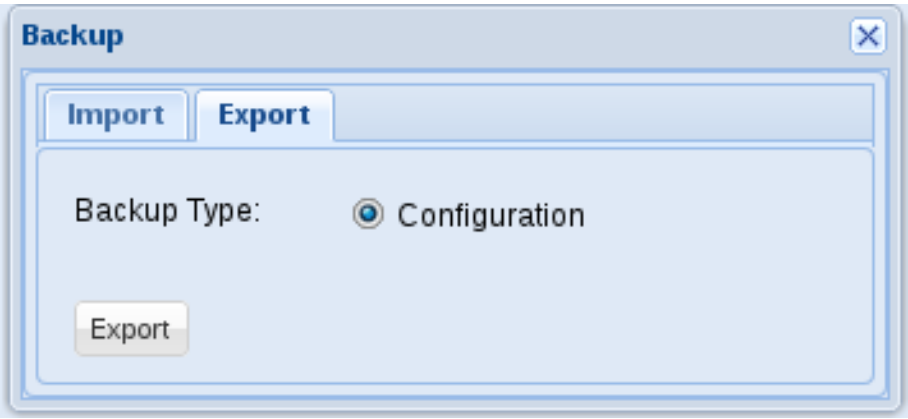

Figure 9.7: Backup export

> The supported phone models are:
>
> - VoiSmart: models Vep-2000, Vep-2100.1, Vep-1000, Vep-1100 and Vep-4000;
> - Snom: models 300, 320, 360 and 370.

To access the autoprovisioning rules, select System → Autoprovisioning Configuration menu and the window to edit current rules or add new one will appear, as shown in figure 9.8 on page 79.

When a rule is added via the *Add* button or edited by double clicking on it, a form with all current available parameters is shown, like in figure 9.9 on page 80. A description of the parameters follows.

***Name*:** a descriptive name of the entry;

***Description*:** notes about the entry;

***SIP Port*:** server SIP port where the device connects to;

***SIP Register Time*:** SIP registration expiry, in seconds;

***Outbound Proxy*:** hostname or IP address of the outbound proxy, if needed. See section 7.3 for a possible usage of the outbound proxy;

***Outbound Proxy Port*:** outbound proxy port contacted by the device;

***Enable Mwi*:** whether to enable MWI on the device;

***Voicemail Number*:** number that must be dialed to reach the voicemail service;

***Server Ntp 1*:** primary NTP server for clock sync;

***Server Ntp 2*:** secondary NTP server for clock sync;

***Display Name*:** by using placeholders, is possibile to select how to show the name of the user on the device. All placeholders must be enclosed between double curly braces. Available placeholders are:

lastname: surname of the user, as configured in the tenant profiles;

firstname: name of the user, as configured in the tenant profiles.

***Call Waiting*:** whether to enable call-waiting on the device;

***WAN Port*:** enable VLAN on the WAN port, assign the ID using the corresponding *Vlan ID* field;

***PC Port*:** enable VLAN on the LAN port, assign the ID using the corresponding *Vlan ID* field;

***Multicast addresses*:** in this section is possible to assign up to ten Multicast listen addresses with their port and label to the phones, which can use to receive Multicast calls. Each address can be called by using the appropriate feature code. Refer to user manual for further details. The format of the address is address:port;

***Firmware URL*:** remote url, with full path, where the phone can download the firmware file. Normally a url, but may vary between phone models.

> ☞ When provisioning a firmware always create separate classes for each phone model, otherwise the phones will try to update them self with the wrong one, which can result into a bricked phone!

Autoprovisioning server

The address of the autoprovisioning server must be configured on the device. The URL is:

- using plain HTTP: http://_ng_hostname_or_ip_address/ap/
- using secure HTTPS: https://_ng_hostname_or_ip_address/ap/

> ✍ Remember the / at the end of the URL!

Several devices support the autoprovisioning server configuration using DHCP. Which DHCP field must be configured with the provided URL depends on the device and on the DHCP server used.

Otherwise is possible to configure the autoprovisioning address directly on the device, if supported.

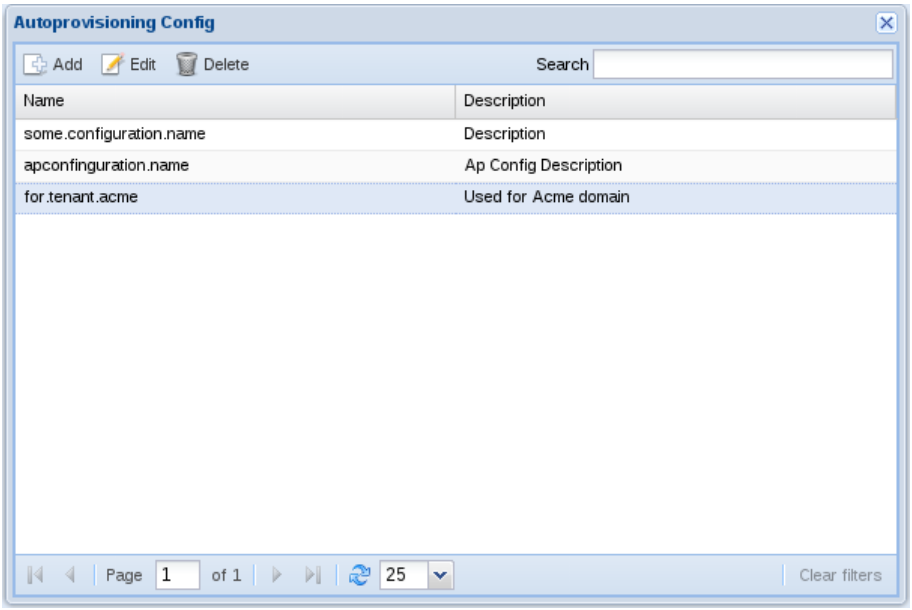

Figure 9.8: Autoprovisioning configurations

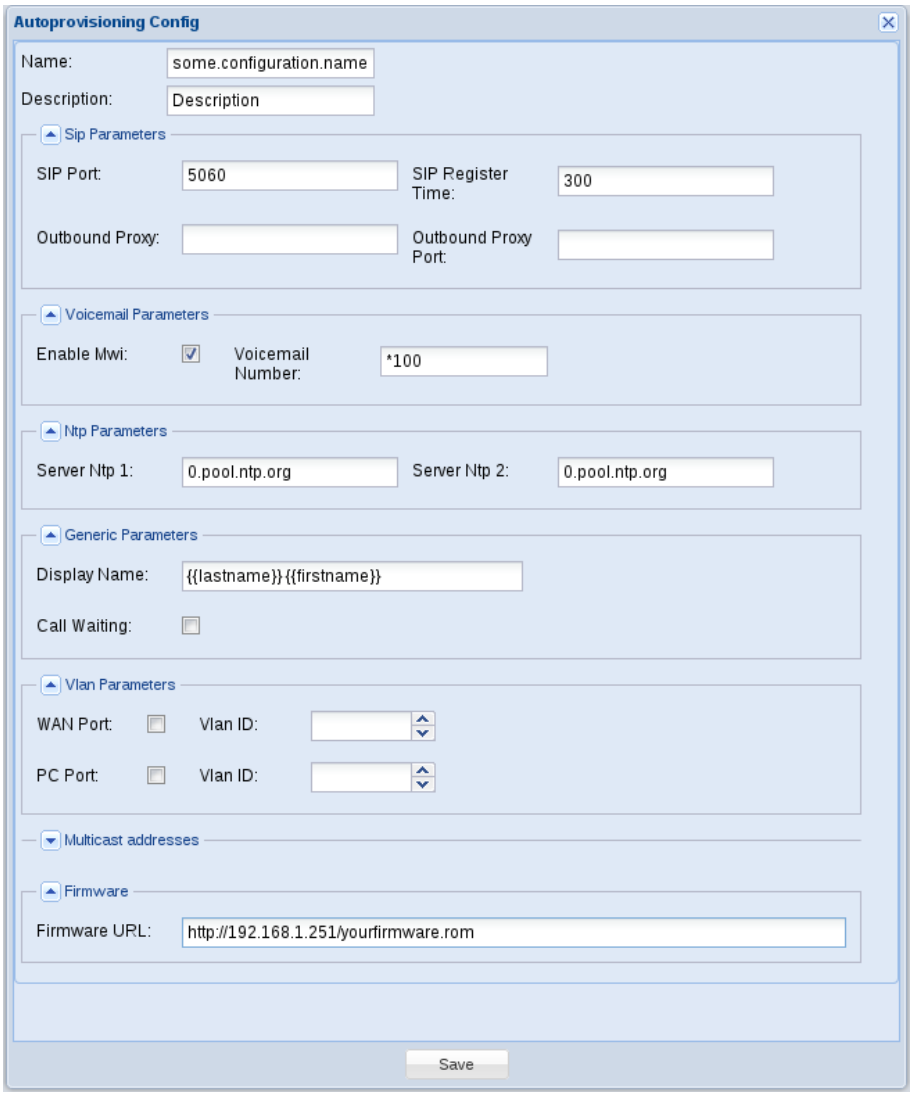

Figure 9.9: Autoprovisioning rule parameters

10

CAC

Contents

10.1 Introduction **82**
10.2 Concepts **82**
10.3 Configuration **82**
10.4 Zones **82**
10.5 Call limits **84**
10.6 Groups **86**
10.7 Putting all together **86**

10.1 Introduction

CAC is a powerful module that allows to partition the devices into zones and apply limits on the number of calls that each device or zone can place at the same time on the Orchestra NG system. This is useful to avoid too may calls if the bandwidth is not sufficient, to limit concurrent calls for each tenant, to simulate a fixed numbers of *lines* like in the PSTN world.

It can also be used to reserve some channels for a specific purpose, like faxing, priority calls (emergency calls).

> ℹ This feature is licensed, you can create or enable items only if you have purchased a cac license.

10.2 Concepts

The basic concepts that needs to be understood before configuration are: *Zones*, *Selectors*, *Limits*, *Groups*.

Zones: a *Zone* is a source or destination of SIP traffic. Is defined by applying one or more traffic *Selectors*;

Selector: a *Selector* is a traffic match rule that applies different criteria on the SIP packets in order to associate them to a specific *Zone*. A *Selector* can match the SIP packet on the SIP domain, source/destination IP network or arbitrary SIP headers;

Limits: a *Limit* represent how many concurrent calls this limit will handle. Each *Limit* has an absolute maximum value (max calls handled by this limit), an outbound value (how many calls can be outbound through the Orchestra NG system) and a borrow limit (how many outbound calls can be borrowed from *Zones* belonging to the same *Group*);

Groups: represent an additional limit value that can be assigned to multiple *Zones* in order to allow them to take more outbound calls when their outbound limit has been reached. Borrowing cannot exceed the absolute maximum limit.

The logical flow of the CAC feature is shown on figure 10.1.

10.3 Configuration

To access CAC configuration select System ⚒ → System Settings and follow the icons under the *Cac* section, described below.

10.4 Zones

By selecting the *Zones* icon the panel shown in figure 10.2 on page 84 will show up. From here it is possible to manage *Zones* definitions and their *Selectors*.

Since *Zones* can sometime overlap, a priority is defined and can be adjusted by dragging the rule and dropping it in the desired position. Only the first matching *Zone* will be used.

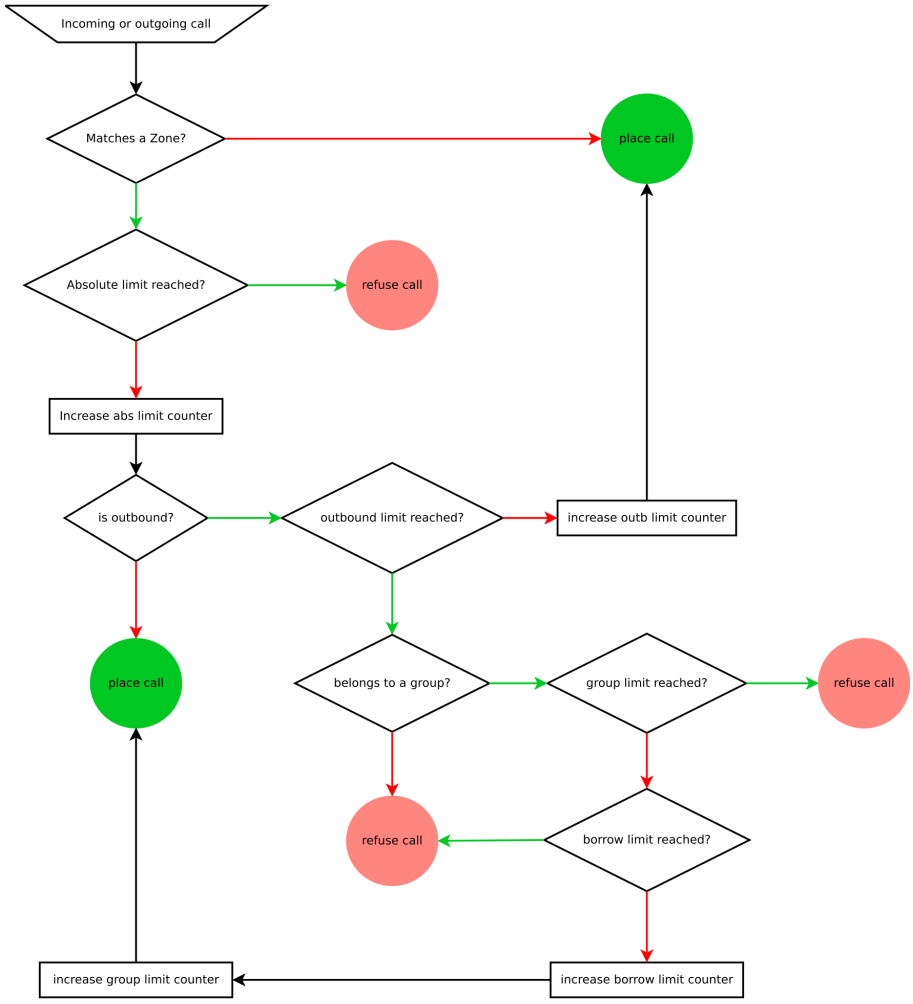

Figure 10.1: CAC logical flow

Zones parameters are:

***Name*:** a descriptive name of the entry;

***Description*:** notes about the entry;

***Selector*:** how to apply the *Zone Selectors*: can be in AND, meaning that all *Selectors* must match, or OR, meaning that one match is sufficient;

By selecting the rule and pressing the *Next* button the *Selectors* panel will show up, as show in figures 10.3 and 10.4. From here the traffic matchers can be created or edited. Each *Selector* analyzes only the SIP INVITE message.

✍ A *Zone* without any *Selector* will not match any traffic.

Selectors parameters description follows:

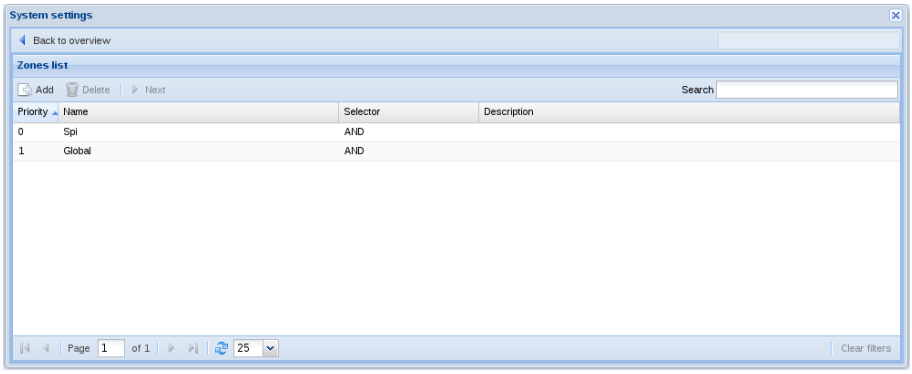

Figure 10.2: CAC Zones panel

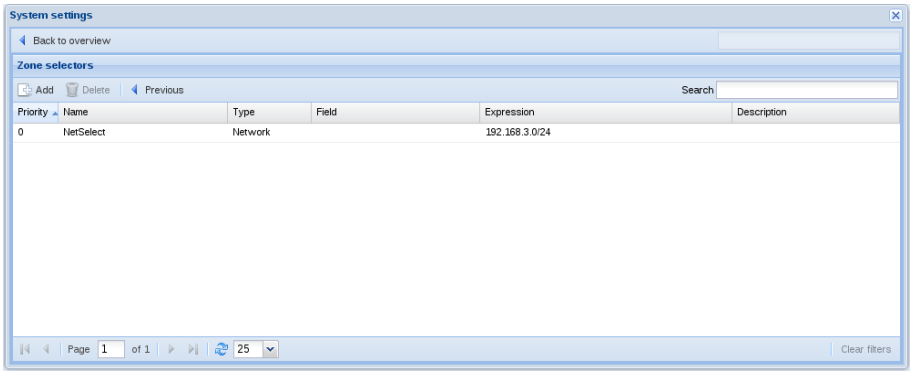

Figure 10.3: Example CAC Selector by network address

***Name*:** a descriptive name of the entry;

***Type*:** type of traffic matcher, can be:

Network: to match against remote IP network address;

Domain: to match against the SIP domain; expression can be a regular expression;

SIP: to match against specific SIP headers that must be specified in the *Field* field; expression can be a regular expression.

***Description*:** notes about the entry.

10.5 Call limits

By selecting the *Call limits* icon the panel shown in figure 10.5 on the facing page will show up. Each limit can have 3 different values, used in different scenarios. A description follows:

***Name*:** a short name for the entry;

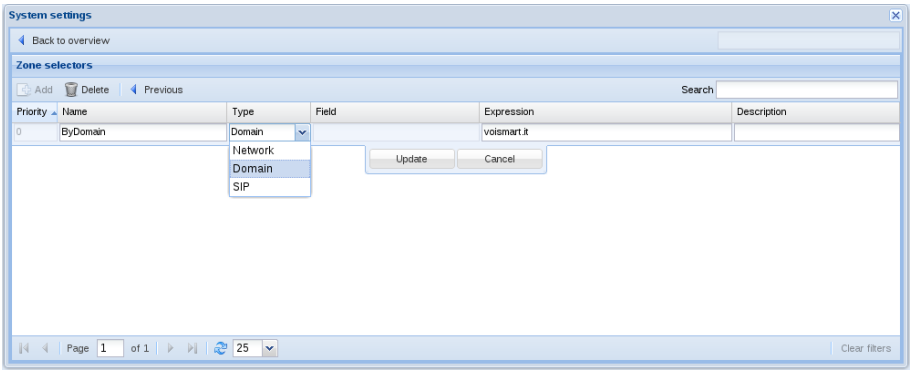

Figure 10.4: Example CAC Selector by SIP domain

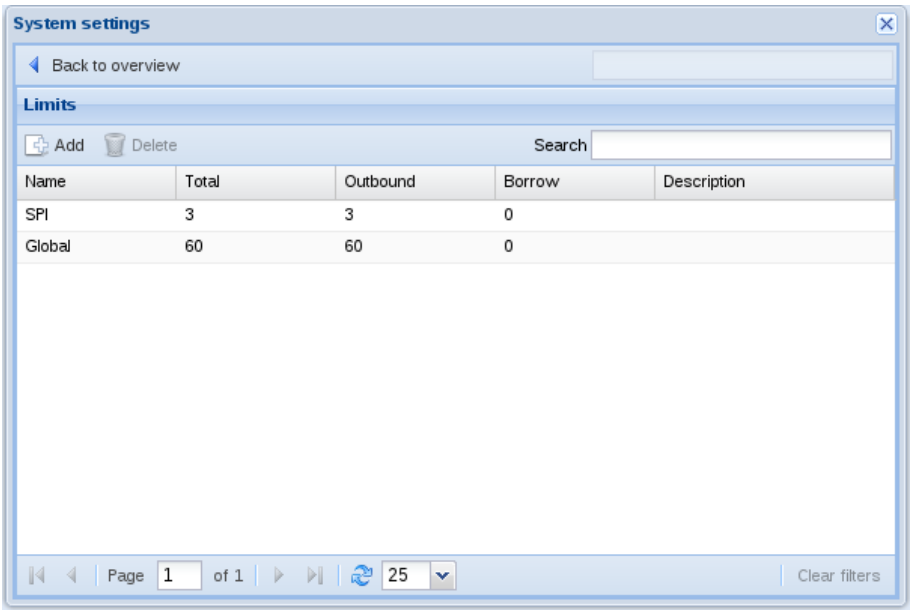

Figure 10.5: CAC Limits panel

***Total*:** absolute maximum concurrent calls limit, a *Zone* with this limit applied will never be able to place more than *Total* concurrent calls;

***Outbound*:** how many concurrent outbound calls can be placed, useful to emulate PSTN fixed-number lines;

***Borrow*:** if the *Zone* with this limit belongs to a *Group*, how many concurrent calls can it borrow from the *Group* pool. The borrow limit from the *Group* pool gets engaged when the outbound limit has been reached;

***Description*:** some notes about the entry.

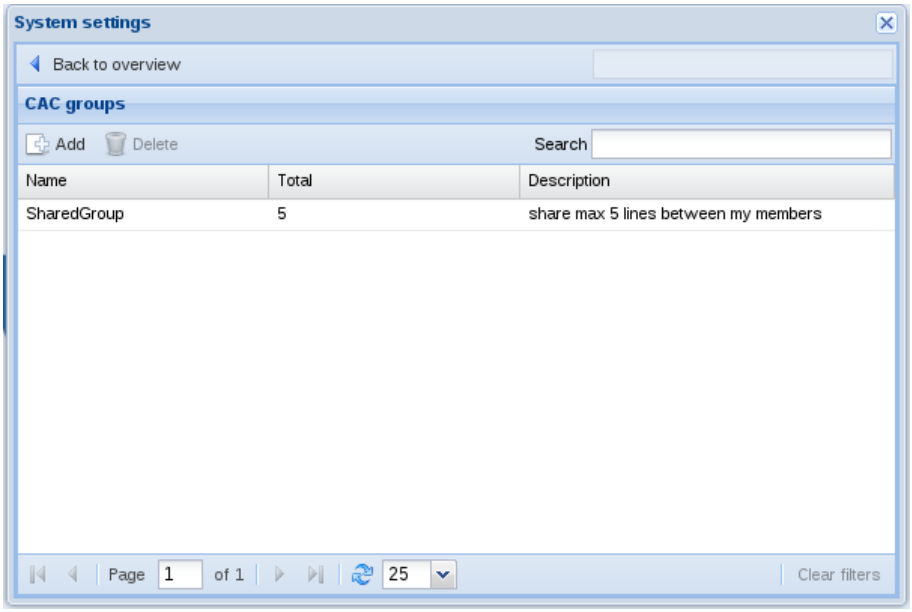

Figure 10.6: CAC Groups panel

10.6 Groups

By selecting the *Cac groups* icon the panel shown in figure 10.6 will show up. *Groups* are used to defined a shared pool of concurrent calls limit that can be borrowed by members of the *Group*, if their outbound limit has been reached. *Group* parameters follows.

***Name*:** a short name for the entry;

***Total*:** maximum number of additional concurrent calls that can be borrowed by members of this rule; each member can borrow up to *borrow* value defined in the corresponding limit;

***Description*:** some notes about the entry.

10.7 Putting all together

By selecting the *CAC* icon the panel shown in figure 10.7 on the facing page will show up. From here is possible to join a *Zone* with a *Limit* and an optional *Group*. This panel is where the real CAC function is activated: if no rule is present here, no concurrent call limits is done, since all previous panels are meant to configure items and here items are associated in a working rule. Each rule has the following parameters:

***Name*:** a short name for the entry;

***Zone*:** select one of the *Zones* configured in section 10.4;

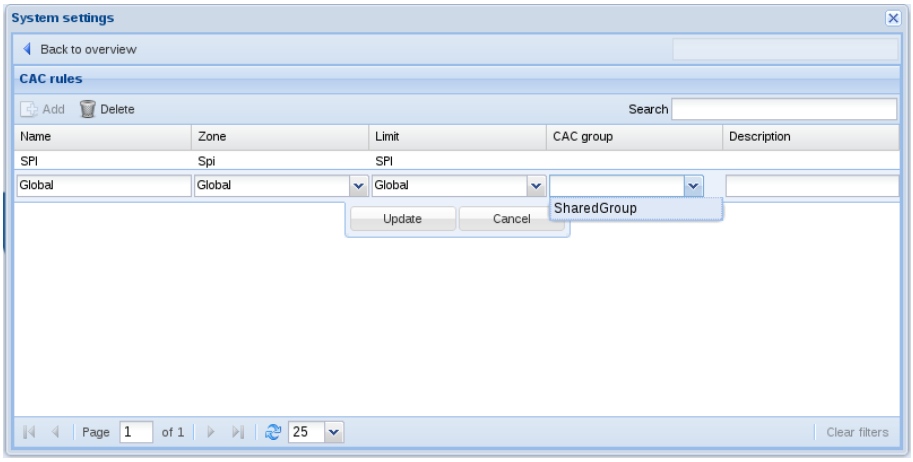

Figure 10.7: CAC Association between Zones, Limits and Groups

***Limit*:** choose which *Limit*, as defined in section 10.5, to apply to the selected *Zone*;

***Cac group*:** select to which *Group* the *Zone* belongs; *Groups* are defined in section 10.6;

***Description*:** some notes about the entry.

11

Security

Contents

11.1 Introduction **90**
11.2 Certificates **90**
11.3 Web GUI **90**
11.4 Firewall **90**
11.5 Accounts **91**
11.6 Backup **92**

11.1 Introduction

The Orchestra NG system is not meant to be a security appliance. While the development of the product takes in account possible misuse and misconfiguration it cannot be 100% secure if proper actions are not done. This chapter wants to indicate common security best practices, but is too far to be exhaustive.

> ☞ Security is not just a set of configurations. Is also a proper knowledge of the deployment, of the technologies involved and of the possible problems. First of all, security is a way of thinking: no manual will ever give you a 100% secure product, if you are not thinking with security in mind.

11.2 Certificates

All Orchestra NG services are available with plain and encrypted connections. To make this possible, the system has an internal Certification Authority which is used to generate certificates for all the services, for all domains (tenants).

Since is not an official Certification Authority, many clients can complain when using the encrypted connections, and commonly is possible to simply *accept* the certificate, by white listing it or ignoring the warning.

To have a more secure and strict certificate check, is possible to export the Certification Authority public certificate in order to add it to the various clients and avoid further warnings.

To export the Certification Authority certificate, visit the URL:

```
http(s)://_ng_hostname_/CA/cacert
```

An X.509 PEM CA root certificate will be downloaded. Refer to the client or operating system manual to figure how to import it as a new, trusted CA .

11.3 Web GUI

The Orchestra NG web gui can be also used with HTTPS. This method is preferred in order to protect the communication between the clients and the server. Since user password are sent in clear text when logging into the system, using SSL prevents information leakage.

11.4 Firewall

The system provides an embedded firewall that is completely managed by the application. If configured from the standard linux commands, it will get reset upon application start. Is always strongly recommended to configure and enable the firewall function. For further information on how to manage it, refer to the administrator manual. If the built-in firewall is not configured, the system will be completely open.

If an external firewall is preferred, the following list describe the ports used by Orchestra NG that must be opened to use the corresponding service.

- 22/TCP: for accessing the local shell, SSL encrypted;

- 25/TCP: for using the internal mail server for mail to fax and related functions;
- 80/TCP: for accessing web gui or provisioning the phones, in clear text;
- 443/TCP: for accessing web gui or provisioning the phones, SSL encrypted. Used also for provisioning of VoiSmart softphones;
- 5222/TCP: used by instant messaging clients;
- 5269/TCP: used by instant messaging server to federate with other XMPP servers;
- 7777/TCP: used by the Hotspot service;
- 16384 to 32768, UDP: for RTP media streams;
- any port defined in the *SIP profiles*, either TCP and UDP.

> ☞ Not all ports must be opened to the public, but only the needed ones. If a service is not to be exposed, do not open the corresponding ports.

11.5 Accounts

One of the common mistakes is to use weak account password. There are two kind of account password in Orchestra NG:

- user accounts: credentials used to connect to the web gui and to provision VoiSmart softphones;
- SIP accounts: used to authenticate SIP phones in order to make and receive calls;
- system accounts: also Linux accounts, the ones that can be used to perform operation from the system shell.

Using weak user password means that a malicious user can connect to the web interface and use or configure service on behalf of another user. If the leaked password belongs to an administrator, the service can be disrupted.

> ☞ One common mistake is to have a weak admin user password, frequently being “admin”, as the default.

Using weak SIP password can lead to use the system as a calling platform for malicious users, resulting in high billing costs. The system already creates a random, secure password for SIP extensions and is suggested to not change it.

> ☞ One common mistake is to have the SIP password equal to the extension which is equal to the SIP username: e.g. extension 200 configured with SIP username 200 and password 200. Common scanners over the public Internet, when an open SIP server is found, try to brute force the SIP accounts by trying all numeric combinations. If one valid account is found, the server is almost

immediately used to place international calls.

Weak system (Linux) accounts can lead to unwanted remote accesses which can transform the system to a remote *zombie* for executing commands; commonly transforming the system into a member of a bigger group of computers using for executing a range of illegal operations (distributed denial of service, IRC bots, storage for illegal files and so on...)

11.6 Backup

The system offers a backup function from the administrative web gui. Always use it and archive in a secure place the exported file. Having it means that is possible to replace a broken unit or a hacked system without incurring into days of downtime. This is more important as the configuration is complex.

✍ Always do a backup before and after a configuration change!

A

DNS MX Records

Orchestra NG services like email-to-fax needs to make the system able to receive emails as a standard SMTP server. When configuring the appropriate services the SMTP service is already active but other mail servers need to be aware how to reach our system.

A.1 How internet mail system works

This appendix is not meant to be a complete guide on the internet mail system, but just provides some pointers for a correct configuration of the environment.

Mail servers are able to reach each other in two principal ways:

- by DNS MX records;
- by direct routing.

Direct routing

This is the simpler way, but only suitable for closed environment. The email domain that is used just needs to be known to the local email servers, where each one knows how to reach the final host with a direct mapping between the email domain and host address. Basically each email client submits its emails to one ore more central servers which have a static mail route that maps the domain to the hostname or IP address of the Orchestra NG system.

This kind of setup needs to configure such mapping on every server that deliver emails for the specific domain. Instructions on how to do this depends on the specific mail server used.

MX record

> ✍ A mail exchanger record (MX record) is a type of resource record in the Domain Name System that specifies a mail server responsible for accepting email messages on behalf of a recipient's domain, and a preference value used to prioritize mail delivery if multiple mail servers are available. The set of MX records of a domain name specifies how email should be routed with the Simple Mail Transfer Protocol (SMTP).

This is the most common setup over the internet, no static routes are needed on any mail server of the network. To make this possible the DNS system is used. First of all, the domain must be valid and resolvable by DNS servers. Then the Orchestra NG system must be published on the DNS servers with a valid hostname. Finally an MX record for the domain must be created and must indicate the Orchestra NG hostname as the mail exchanger for the specific domain.

☞ With the latter method, the Orchestra NG system needs to be reachable by all the hosts participating to the mail exchange: if the domain is routed globally, the host must be globally reachable.

B

CSV Export cdr

Example CSV file for export cdr:

```
id,direction,name,user_name,caller_id,domain_name,billusec,created_time,answered_time,
    ↪ hangup_time,hangup_cause,destination_number,endpoint_disposition
976,inbound,,milleuno,"""milleuno"" <milleuno
    ↪ >",192.168.3.95,6539454,1411388240992190,1411388243612343,1411388250151797,
    ↪ NORMAL_CLEARING,1000,ANSWER
977,inbound,,1000,"""vm""
    ↪ <1000>",192.168.3.95,2039596,1411389653772154,1411389653812144,1411389857771820,
    ↪ NORMAL_CLEARING,*100,ANSWER
978,outbound,,,"""Bobby Fischer"" <milleuno>",,0,1411389903411310,0,1411389916291738,
    ↪ USER_BUSY,1000,
979,inbound,,milleuno,"""milleuno"" <milleuno
    ↪ >",192.168.3.95,39819510,1411389902571226,1411389916451745,1411389956271255,
    ↪ NORMAL_CLEARING,1000,ANSWER
980,outbound,,,"""Bobby Fischer"" <milleuno>",,0,1411389997832090,0,1411390001011638,
    ↪ USER_BUSY,1000,
981,inbound,,milleuno,"""milleuno"" <milleuno
    ↪ >",192.168.3.95,9520212,1411389997291282,1411390001191648,1411390010711860,
    ↪ NORMAL_CLEARING,1000,ANSWER
982,inbound,,1000,"""vm""
    ↪ <1000>",192.168.3.95,2608108,1411390042651076,1411390042711095,1411390068792175,
    ↪ NORMAL_CLEARING,*100,ANSWER
983,outbound,,,"""Bobby Fischer"" <milleuno>",,0,1411390092991757,0,1411390095290965,
    ↪ USER_BUSY,1000,
984,inbound,,milleuno,"""milleuno"" <milleuno
    ↪ >",192.168.3.95,31699733,1411390092551383,1411390095591366,1411390127291099,
    ↪ NORMAL_CLEARING,1000,ANSWER
```

Domain_name field is available only when you export records using system administration web interface.

C

Regular Expressions

This section is not meant as a definitive guide for *Regular Expressions*, but as small introduction with commonly used patterns.

A *Regular Expression*, called also *regexp*, is a string expression that describe a set of strings that matches it. With a regexp it is possible to create a rule that matches several input strings, like numbers, thus avoiding the definition of many match rules. A single match task can frequently be specified by different expressions, so there's not a single way to obtain the same result.

> ✌ If it is needed to match all numbers in the range `123400-123499`, instead of writing 100 single matches (`123400`, `123401`, ..., `123499`) it is possible to define a rule like `1234..` that matches all the range.

If a literal match against a symbol is needed, escape it with \. For example to match an asterisk (`*`) in a number, the expression `\*` must be used.

On table C.1 all supported symbols are shown. On table C.2 some application examples are reported and refer to table C.3 for some substitution examples.

Table C.1: Regular expressions valid symbols.

Symbol	Description
`.`	Indicates a single-digit placeholder. For example, `1234...` matches any dialed number beginning with `1234`, plus three additional digits.
`*`	Indicates that the preceding digit or pattern occurred zero or more times.
`+`	Indicates that the preceding digit or pattern occurred one or more times.
`?`	Indicates that the preceding digit or pattern occurred one or zero times.
`|`	*OR* operator. If `A` and `B` are regular expressions, `A|B` will match any string that matches either `A` or `B`.
`{m,n}`	Indicates that the preceding digit or pattern occurred at least `m` and at most `n` times. `n` can be omitted and is assumed to be infinity if using a trailing comma, `m` otherwise.

Table C.1: Regular expressions valid symbols.

Symbol	Description
`( )`	Indicates a group of patterns, also can be repeated with a repeating qualifier, such as `*`, `+`, `?`, or `{m,n}` For example, `(ab)*` will match zero or more repetitions of `ab`. Groups are also captured and can be used in replace rules by referring to them with the `$idx` or `{idx}` syntax, where `idx` is the index of the matched group, starting from `1`.
`[ ]`	Indicates a character class, which is a set of characters that you wish to match. Characters can be listed individually, or a range of characters can be indicated by giving two characters and separating them by a `-`. For example, `[abc]` will match any of the characters `a`, `b`, or `c`; this is the same as `[a-c]`, which uses a range to express the same set of characters. `[09]` matches only `0` or `9` while `[0-9]` matches all ten digits from `0` to `9`. To match only all lowercase letters, your expression would be `[a-z]` Symbols are not active inside classes, `[a2*]` matches `a`, `2` and `*` characters. `^` as first character in a class is a complementing set, which indicates to match everything except the set. For example `[^0]` matches everything except `0`.
`\d`	Matches any decimal digit; this is equivalent to the class `[0-9]`
`\D`	Matches any non-digit character; this is equivalent to the class `[^0-9]`
`\s`	Matches any whitespace character; this is equivalent to the class `[\t\n\r\f\v]`
`\S`	Matches any non-whitespace character; this is equivalent to the class `[^\t\n\r\f\v]`
`\w`	Matches any alphanumeric character; this is equivalent to the class `[a-zA-Z0-9_]`
`\W`	Matches any non-alphanumeric character; this is equivalent to the class `[^a-zA-Z0-9_]`

Table C.2: Regular expressions examples.

Expression	Description
`0?12345678`	Matches the number `12345678` with an optional `0` at the beginning
`021234..`	Matches the numbers from `02123400` to `02123499`

Table C.2: Regular expressions examples.

Expression	Description
`021234[0-9]{2}`	Same as above
`0212[3-6]\d+`	Matches from `02123` to `02126` followed by one or more digits, so `021231`, `02123544`, `0212400` all are possible matches
`0212[3-6].+`	Similar to above, but will match also all characters after `[3-6]`, not only digits
`445566+`	Matches `44556` followed by one or more `6`, like `445566`, `4455666`, `4455666666`, ...

☞ Avoid match-all rules like `.*` because it can lead to unwanted side effects. Always use as strict as possible matching patterns. Laziness is not a good reason to use match all expressions.

Table C.3: Regular expressions substitutions.

Expression	Replacement	Description
`0?12345678`	`1234`	`12345678` with an optional `0` at the beginning is blindly rewritten to `1234`
`021234(..)`	`456$1`	numbers from `02123400` to `02123499` are rewritten to `45600` to `45699`
`021234(..)`	`456$10`	Error! Group `10` does not exists!
`021234(..)`	`456${1}0`	numbers from `02123400` to `02123499` are rewritten to `456000` to `456990`
`02(\d{3})([0-9][56])`	`${2}55${1}`	numbers like `02 123 45` are rewritten to `45 55 123` (note where the groups expansion is used)

D

Protocols and standards

> ✍ The following list of supported protocols and standards is provided for information purposes only and is not meant to be complete or guarantee of compatibility with other devices or implementations. Some protocols or standards may not be fully implemented.

D.1 VoIP

Signalling

- UDP, TCP and TLS transports
- RFC 2617: HTTP Digest Authentication
- RFC 3261: SIP v2.0
- RFC 3262: PRACK and 100rel
- RFC 3263: Locating SIP Servers
- RFC 3264: SDP Offer/Answer Negotiation
- RFC 3265: SIP Event Notifications
- RFC 3323: Privacy
- RFC 3325: Asserted Identity
- RFC 3327: Path
- RFC 3515: REFER
- RFC 3551: RTP/AVP
- RFC 3711: SRTP
- RFC 3842: Message waiting event
- RFC 3856: Presence
- RFC 3892: Referred-By
- RFC 3891: Replaces
- RFC 4028: Session Timers
- RFC 4566: SDP Session Description Protocol

Media

- G.711u, G.711a
- G.722, G.722.1
- G.729: requires a separate channels license
- iLBC
- Speex
- H.263 (pass through only)
- H.263-1998 (pass through only)
- H.263-2000 (pass through only)
- H.264 (pass through only)
- T.38: fax over IP networks in real time

D.2 Fax

- V.21
- V.27ter
- V.29
- V.17
- ECM (error correcting mode)
- T.4 1D, T.4 2D, and T.6 image compression

D.3 PSTN Telephony

- FXS, FXO: analog telephone signalling and interfaces
- BRI, PRI: basic rate and primary rate digital interfaces
- ISDN: digital transmission over PSTN
- Q.921
- Q.931: signalling for DSS1

D.4 Others

- HTTP, HTTPS: web GUI and API access
- RFC 6455: Websockets
- SSH: remote management
- XMPP: instant messaging
- SSL: encryption of several communication channels, like HTTP, VPN, IM, SIP, ...

Glossary

ACD In telephony, an automatic call distributor (ACD) or automated call distribution system, is a device or system that distributes incoming calls to a specific group of agents based on customer need, type, and agent skill set.

ALG Application-level gateway. In the context of computer networking, an application-level gateway consists of a security component that augments a firewall or NAT employed in a computer network. It allows customized NAT traversal filters to be plugged into the gateway to support address and port translation for certain application layer "control/data" protocols such as SIP and others. 54

ATA An analog telephony adapter or analog telephone adapter (ATA) is a device used to connect one or more standard analog telephones to a digital telephone system, such as voice over IP. 49

BRI A Basic Rate Interface, is one of the kinds of ISDNaccess interfaces. A BRI provides 2 B-Channels, for transporting data, and a single D-Channel, for signalling and control. 44–46

CDR Call detail record, is a collection of data records produced as a result of phone calls and including details such as dialed number, call duration, call creation date and time, ... Used for billing or logging purposes. 24

DSS1 Digital Subscriber Signalling System No. 1 (DSS1) is a digital signalling protocol (D channel protocol) used for the ISDN. It is defined by ITU-T I.411 (ETS 300 102). It supports Bearer Capability, Low Level Compatibility and High Level Compatibility, ANI, DNIS and redirected number signaling in both directions. A standard developed by ETSI for Europe is known as Euro-ISDN or E-DSS1 or simply EDSS1 (European DSS1). See also ISDN. 46, 47, 103, 104

E1 An ITU-T and ETSI standard designed to carry digital signals over telephony trunks or lines at a bit rate of 2048 Mbit/s. Commonly used in Europe.

E.164 E.164 is an ITU-T standard which defines a numbering plan and general format of numbers for telephony systems. 25

Euro-ISDN see DSS1. 44, 45

FDR Fax detail record, is a collection of data records produced as a result of fax calls and including details such as dialed number, call duration, transmission result, ... Used for billing or logging purposes. 24

FXO A Foreign eXchange Office is an interface port that receives an analog line from the PSTN. An FXO device is a device with an FXO port attached, such as a phone or a fax. An FXO gateway is a device providing interconnection between an analog line and an IP-PBX. 44, 45

FXS A Foreign eXchange Subscriber is an interface port that delivers an analog line to the subscriber. An FXS gateway is a device providing interconnection between an IP-PBX and an FXO port. 44, 45

ISDN Integrated Services for Digital Network, is an ITU-T and ETSI standard to allow transmission of data and voice over a PSTN network. It is also common to refer to ISDN as the circuit-switched physical network itself. See also DSS1. , 103

span A span represents a single physical port of any TDM interface. 44, 45, 47

LCR LCR, or Least-cost routing, is the functionality of selecting the route for an outbound call based on some configurable settings, generally based on cost, but also on time and date, dialed number and so on. Can also mean Least-cost router, a single system component which routes the call according to a least-cost rule. 25, 50, 58, 61

LDAP Lightweight Directory Access Protocol, is a protocol implemented by applications providing directory services, i.e. a software which provides to clients informations about users and services throughout the network. 36, 38

NAT Network Address Translation is the task of rewriting IPdata packets headers when leaving a network element such as a router or gateway, with the purpose of IP masquerading or more generally mapping a set of IP addresses into another (e.g. for hiding a private LAN address space behind a single address, usually in the public space).

PRI A Primary Rate Interface, is one of the kinds of ISDN access interfaces. A PRI provides 30 E1 B-Channels or 23 T1 B-Channels. 44, 46

PSTN The Public Switched Telephone Network, is the infrastructure providing public telephony and telecommunications, and is the collection of the interconnected circuit-switching networks around the globe and operated by telephony providers. 44

queue A queue is a simply ordered list of calls to be dispatched to agents. The algorithm by which calls are dispatched is called the queue strategy. , 109

RTP The Real-time Transport Protocol is a protocol used to carry media streams over IP networks and it is often used in conjunction with the SIP signalling protocol. 91

SBC A Session Boarder Controller is an entity used in VoIP networks which acts as a controller over the signalling and media of VoIP calls, usually placed on the edges of different networks to provide different services such as securing the internal network from the outside, media transcoding, signalling protocol translations, topology hiding, applying QOS policies, ...

SIP The Session Initiation Protocol is an IETF-standardized signalling protocol used to control media sessions like voice or video calls. 36, 38, 72

SIP proxy A SIP proxy server is a SIP client and server software which acts as an intermediary between other SIP entities providing features such as routing, rewriting, applying policies and interpreting received messages before forwarding to a SIP server.

SIP trunk The term "trunk" derives from its use within circuit-switched telephony systems and, in the context of SIP, usually means a virtual sip entity on a server which process a request according to a predefined set of polices and rules. This is usually regarded as the VoIP equivalent of a trunk in the TDM world and can be used as a connection beetween SIP servers to provide inter-domain communication, to provide PSTN termination by connecting to a gateway service or an ITSP, and so on. 25, 53

SIP gateway A *SIP gateway* in Orchestra NG is a SIP trunk which provides a connection to an ITSP, another Orchestra NG instance or a generic SIP-compliant IP-PBX.

SIP profile A *SIP profile* in Orchestra NG, is a set of common configurations to be applied to media calls, used to be able treat differently devices connected to different network segments. A *SIP profile* is commonly identified by a unique combination of the IP-port pair.

TCP The Transmission Control Protocol is a transport layer protocol providing an ordered, reliable and error-checked transmission of data packets over a network. 90, 91

TDM TDM, or Time-division-multiplexing, is a form of signal multiplexing (i.e. a way of conveying multiple signals or bit-streams on a shared communication medium) used for call interleaving in telephony systems. 25, 44, 104

AES The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

B-channel A *bearer* channel in an ISDN network providing data and voice transport at a *full-duplex bit rate* of 64 Kbit/s.

BLF a Busy Lamp Field is a visible indicator, usually a light or led, which shows the status of another terminal connected to the same PBX.

CAC The Call Admission Control is a preventive procedure to control traffic congestion by limiting and possibly rejecting a call, according to some configured rules. 25, 50, 74, 82, 86

CSV A Comma Separated Values (also sometimes called Character-Separated Values, because the separator character does not have to be a comma) file stores tabular data (numbers and text) in plain-text form. 95

Carrier A carrier in Orchestra NG is a logical group of trunks that can be associated to an LCR rule. 25, 58

Codec Short for Coder-Decoder, a codec is a software or hardware-based program designed to encode and decode a signal. 50

D-channel A *delta* channel in an ISDN network providing signalling and control information at a 16 Kbit/s rate for BRI, and 64 Kbit/s for PRI.

DHCP DHCP is an IP network protocol used to automatically assign network configurations such as DNS server, the default gateway, . . . to clients.

DNS DNS is a service which translates domain names into IP addresses. 38, 94

DSS Direct Station Select, is the feature of having a group of keys on a terminal to select other terminals or stations to call. Often associated with a BLF indicator.

DTMF Dual-tone multi-frequency signaling (DTMF) is an in-band telecommunication signaling system using the voice-frequency band over telephone lines between telephone equipment and other communications devices and switching centers.

Extension An extension refers to a phone (physical or software-based) connected and configured on an IP-PBX.

Feature Codes Feature Codes are codes allowing you to use the dial-pad on your telephone to access, activate, deactivate special features on an IP-PBX.

HTML5 HTML5 is a core technology markup language of the Internet used for structuring and presenting content for the World Wide Web. As of October 2014 [update] this is the final and complete fifth revision of the HTML standard of the World Wide Web Consortium (W3C).

Hotspot A hotspot is a site that offers Internet access over a wireless local area network (WLAN) through the use of a router connected to a link to an Internet service provider. Hotspots typically use Wi-Fitechnology.

IMAP The Internet Message Access Protocol is an IP protocol used for email retrieval. When used over SSL-secured connections, it is often used the term IMAPS.

IM Instant messaging (IM) is a type of online chat which offers real-time text transmission over the Internet or any other IP network. Short messages are typically transmitted bi-directionally between two parties, when each user chooses to complete a thought and select "send". Some IM applications can use push technology to provide real-time text, which transmits messages character by character, as they are composed. More advanced instant messaging can add file transfer, clickable hyperlinks, Voice over IP, or video chat.

IVR In telecommunications, IVR allows customers to interact with a company's host system via a telephone keypad or by speech recognition, after which they can service their own inquiries by following the IVR dialogue. IVR systems can respond with prerecorded or dynamically generated audio to further direct users on how to proceed. IVR applications can be used to control almost any function where the interface can be broken down into a series of simple interactions.

MP3 MPEG-1 or MPEG-2 Audio Layer III, more commonly referred to as MP3, is an audio coding format for digital audio which uses a form of lossy data compression.

MWI A message-waiting indicator (MWI) is a telephone feature that illuminates a generic indicator like a LED or icon on the LCD display, to notify the user of waiting voicemail messages on the IP-PBX. 77

Multicast In computer networking, multicast (one-to-many or many-to-many distribution) is group communication where information is addressed to a group of destination computers simultaneously. 78

NAS A network access server (NAS) is a single point of access to a remote resource. It is meant to act as a gateway to guard access to a protected resource. This can be anything from a telephone network, to printers, to the Internet.

NTP Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. 77

POP The Post Office Protocol is an IP protocol used for email retrieval. When used over SSL-secured connections, it is often used the term POPS.

QOS The Quality of Service is used to guarantee a certain level of performance in telephony and computer networks by providing the ability to limit or assign different priorities to single applications, users, type of traffic and so on. Common requirements are often fixed bit rates, response delays, jitter...

RADIUS Remote Authentication Dial In User Service (RADIUS) is a IETF standarized networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for users that connect and use a network service.

Realm A Realm is the set of authentication resources, usually consisting of an authentication server and its security policies.

Roster In XMPP, one's contact list is called a roster, which consists of any number of specific roster items, each roster item being identified by a unique JID (usually of the form <contact@domain>). A user's roster is stored by the user's server on the user's behalf so that the user may access roster information from any resource.

SMTP The Simple Mail Transfer Protocol is an IP protocol used for email transmission. When used over SSL-secured connections, it is often used the term SMTPS. 25, 70, 93

SRV record An SRV record is a DNS record type used to define the location on the network of specific services such as SIP or XMPP servers.

SSL The Secure Sockets Layer (SSL) is a cryptographic protocol for secure communication and providing strong encryption of network traffic at application level protocol. , 108

TLS The Transport Layer Security is a cryptographic protocol for secure communication and providing strong encryption of network traffic for application level protocols like HTTP, SMTP, ... 50, 70

TON TON or type of number, indicates the scope of the address value, such as whether it is an international number, a national number, unknown or other formats. 46–48

Trunk In the TDM world, a trunk is a physical line or circuit connecting telephony switches, providing a transmission channel between two elements. In this document, the term TDM will be used interchangeably to denote a TDM trunk or a SIP trunk, as it is common practice in the telecommunication field. 25, 36, 44, 58

VLAN In computer networking, a single layer-2 network may be partitioned to create multiple distinct broadcast domains, which are mutually isolated so that packets can only pass between them via one or more routers; such a domain is referred to as a virtual local area network, virtual LAN or VLAN. 78

VPN A Virtual Private Network is a *point-to-point* connection used to link two private networks over a public one (such as the Internet), using a secure, encrypted tunnel. 25

WAV Waveform Audio File Format (WAVE, or more commonly known as WAV due to its filename extension) (rarely, Audio for Windows) is a Microsoft and IBM audio file format standard for storing an audio bitstream on computers.

WSS WSS stands for secured WebSocket, a WebSocket over SSL. 50

WebRTC WebRTC (Web Real-Time Communication) is an API definition drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling, video chat, and P2P file sharing without plugins. 50

WebSocket WebSocket (WS) is a protocol providing full-duplex communications channels over a single TCP connection. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011. , 108

XMPP Extensible Messaging and Presence Protocol (XMPP) is a communications protocol for message-oriented middleware based on XML (Extensible Markup Language). The protocol was originally named Jabber, and was developed for near real-time, instant messaging (IM), presence information, and contact list maintenance. Designed to be extensible, the protocol has also been used for publish-subscribe systems, signalling for VoIP, video, file transfer, gaming, Internet of Things (IoT) applications such as the smart grid, and social networking services. Unlike most instant messaging protocols, XMPP is defined in an open standard and uses an open systems approach of development and application, by which anyone may implement an XMPP service and interoperate with other organizations' implementations.

bricked The word "brick", when used in reference to consumer electronics, describes an electronic device such as a phone, router, router, or tablet computer that, due to a serious misconfiguration, corrupted firmware, or a hardware problem, can no longer function, hence, is as useful as a "brick". 78

T1 An ITU-T and ETSI standard designed to carry digital signals over telephony trunks or lines at a bit rate of 1544 Mbit/s. Commonly used in North America and Japan.

UDP The User Datagram Protocol is a transport layer protocol providing a very simple model for packet transmission over a network without guarantees on order, reliability or retransmission of lost data. 91

Wi-Fi Wi-Fi, also spelled Wifi or WiFi, is a local area wireless technology that allows an electronic device to exchange data or connect to the internet using 2.4 GHz UHF and 5 GHz SHF radio waves. 25

Agent In Orchestra NG, an agent is a user who is a member of one or more queues.

Autoprovisioning Autoprovisioning is the process of auto configuring IP-phones via a central configuration server like Orchestra NG.

Domain In Orchestra NG, a domain is a single tenant which organizes togheter related user and extensions in a way that its configuration (dialplan, extensions, call routing, service classes, . . .) is completely isolated from other users on the system. That level of separation is granted by SIP domains. Usually in Orchestra NG, a different domain is created for every company o tenant which uses the system. 25, 36, 38

RPM Red Hat Package Manager or RPM Package Manager (RPM) is a package management system. The name RPM variously refers to the .rpm file format, files in this format, software packaged in such files, and the package

manager itself. RPM was intended primarily for Linux distributions; the file format is the baseline package format of the Linux Standard Base. 21

Station ID A station ID is a short string (typically less than forty characters) which identifies the fax machine, and is printed as an header on received and sent documents. Usually denoted as "Called subscriber identification" or "Transmitting subscriber identification" when differentiating the receiving and trasmitting end.

yum The Yellowdog Updater, Modified (yum) is an open-source command-line package-management utility for Linux operating systems using the RPM Package Manager. 21