OpenPhonyx

PBX for real world

Lukas Macura
Silesian University
macura@opf.slu.cz

ABSTRACT

Today, there are many multimedia systems around the world. Most of VoIP-Only systems are multimedia ready. Some of them are configurable using config files, some are configured using some GUI. This article will discuss about command-line interface which should be robust and easy. Entire system shiould be configurable using one CLI. We should not edit any file by hand to make entire system working.

Keywords

VoIP, SIP, H.323, IAX, ISDN, Asterisk, Embedding, Phonyx, OpenPhonyx

1. INTRODUCTION

New config system should be robust, easy and clear to read. We need hierarchical config with more contexts. Variables and features should be inherited from upper level tree to lower. Eg. global parameter for entire system should be inherited into all subtrees until not changed. Entire system explained in graph.

2. CONTAINER MODEL

See figure 1 for overall lookup of containers and hierarchy. Container is group of setting which are glued together and are hierarchical. Parameters are inherited from upper level container into lower level

2.1 System

System is upper level container. All containers are inherited from from it. It can contain one or more servers. All of them will share one config database. It means, for operator, system is entire VoIP infrastructure, while server is one of physical PCs which are serving database.

2.2 Server

Server is physical PC which is serving mulimedia application. It is connected to System and sharing most of settings with other servers in same system. Server can be primary or backup device for PBX. One organisation will use one system, which can be managed by one CLI, but can have more servers, to do better load balancing, failover and geografical server placement.

2.3 PBX

One of PBX which will be on system. Entire communication system should be multi-PBX. Can be real, virtual or backup type. Real means this system will native serve it as primary server. Backup means it will server this PBX just after primary server is down. Virtual means this is not real PBX for registering users but helper for callrouting etc. One PBX can ”include” other PBXes. They are hierarchical.

2.4 Pricelist

Pricelist are used for accounting, billing and prepaid. There are two kind of prices - operator and user and two kind of directions - input and output. PriceList supports more types of tarifications. (1S/1S, 1S/30S, 1S/infinity, and totaly custom). Operator prices are used for statistics, user prices are used for user accountings. Prices are bound to PBXes. One PBX can have only one input pricelist, operator or user type and one output, respectively. Pricelist can be assigned to special actions too. Eg. autoconfig SMS will be accounted using SMS pricelist.

2.5 Admin

Admins of the system. Admins are absolutely another kind of users. They are managing system, not using the system for calling. All logic is strictly divided to user and admin part. User can change their userparms and lineparms, but not other things.

2.6 AdminParms

Parameters for admin of the system. Default PBX, servers and settings of the CLI for admin are stored here.

2.7 User

User which use system. Can be employee or customer. One user can have more phones associated. User is used for configuring his own phone and line parameters, accessing web gui and billing purposes. Each user has userparms associated.

2.8 UserParms

User parameters. It means global setting for lineparms, which are inherited for each new user line. There are another things like user PIN, user email, name, visibility in phonebook, etc. Userparms are extensible, there can be huge number of them per user.

2.9 UserGroup

Group of users. User can be in one or more group. Group are used for reporting, rights or simplified config. Not for billing and accounting.

2.10 Number

Means free number which is in dialplan but is not associated or used. Each number belongs to some area. Area is feature of number, not line. Number cannot have more areas. There are two types of numbers. Globaly unique and rest. Globaly unique has to be in e.164 form. They have to be unique for entire system. Rest of numbers are localy unique. It means unique in PBX. Similary to NAT, localy unique numbers are not accessible from outside without specialised redirects or IVR.

2.11 Line

Line is number, associated to user, it means active number. Line can have more types and more protocols. It means, SIP PBX line, SIP operator line, IAX2 PBX line, IVR menu etc. Each line has lineparms associated. Line can be part of more groups, but only one Area. Lines has to be unique per server (including backup).

2.12 Phone

Phone can be connected to line. Line can have zero ore one phone connected. If line is not connected to phone, user has to setup it’s own software or hardware by own. If there is connected phone with known type, system will make autoconfig and provision for it.

2.13 PhoneParms

Phone parameters. Can be mac, IP, serialno, state, ...

2.14 LineGroup

Group of lines. Can be used for some reporting, rights or simplified config. Not for billing and accounting. Line can have one or more groups associated.

2.15 LineParms

Line parameters. In most situations, these are parameters which can be edited by user and are not system specific. Lineparms are extensible, there can be huge number of them per line. Examples:

- CFU
- CFB
- CFNR
- CFNRTimeout
- NRNotify

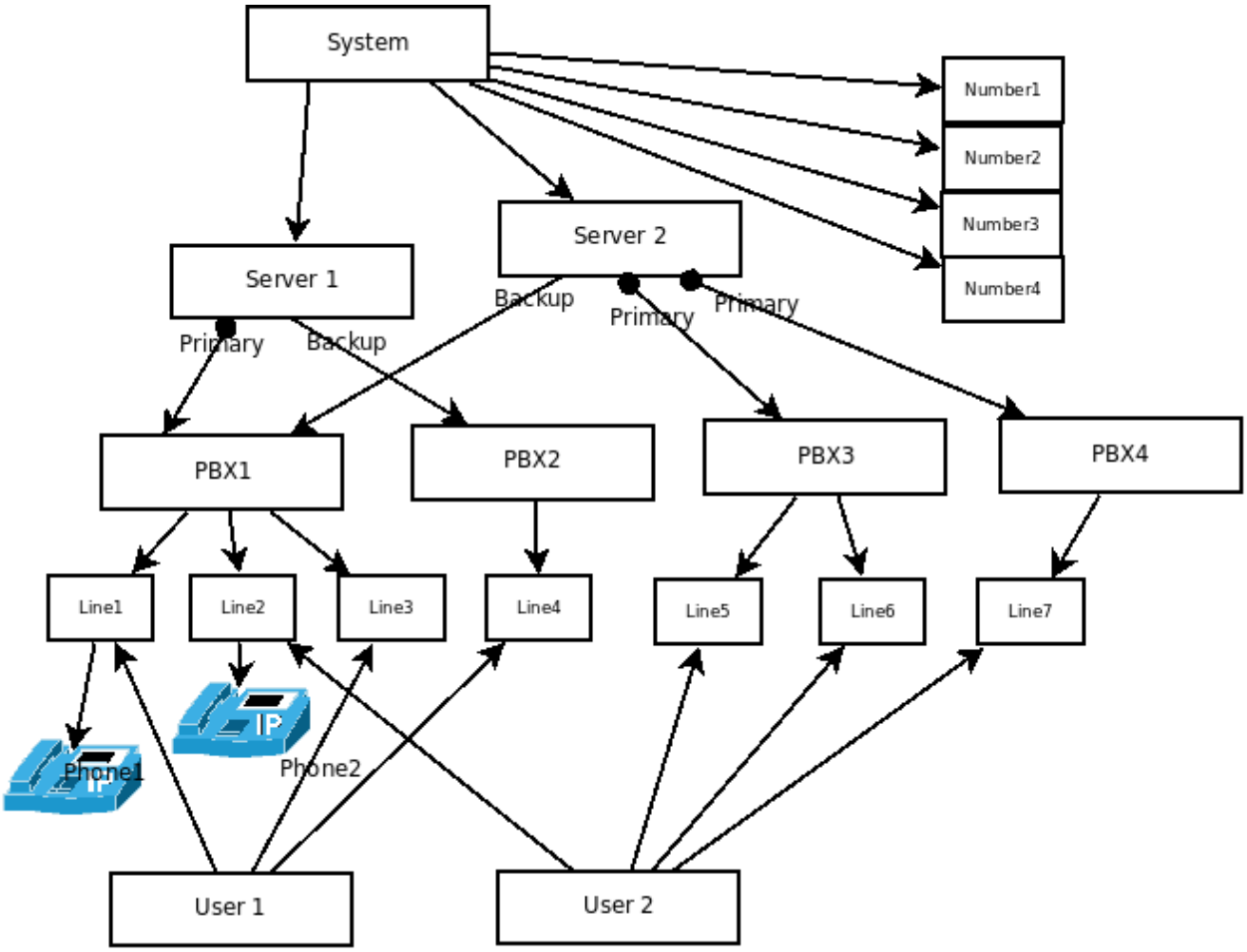

Figure 1: Container hierarchy

- BNotify
- UNotify
- VoiceMail
- Monitoring
- Email

2.16 Area

Area of numbers. For example, it can be prefix of numbers which are on one building. Or, it can be prefix of lines used in some area (Czech Republic) for operator. In most situations, areas are divided from line format. Eg. area "Czech Republic" are globaly unique numbers starting +420.

3. SEMANTICS

Basic semantics and usage of CLI, divided to containers and their functions. There are predefined commands which are same for all containers.

3.1 All Containers commands

Commands which are same for all containers. In general, when config is changed by some command, no actions are taken, only semantics check of commands. To apply changes, there is apply command. To revert them back, there is revert command.

3.2 All Containers return messages

Each line should return number and text. Numbers and texts are similar to HTTP retun codes.

3.2.1 apply

Listing 1: Apply options

```
Syntax: apply [late]
Example:
>line add 111 type ivr
202 OK
>line add 112 type ext protocol sip
202 OK
>pbx modify users add line 111
202 OK
>pbx modify users add line 112
202 OK
>apply
200 Applied
```

Will save changes to DB and apply changes made to containers. If optional parameter late is entered, will not apply now but when system is idle (there are no calls). See listing 1.

3.2.2 revert

See listing 2.

Listing 2: Revert options

```
Syntax: revert
>line add 111 type ivr
202 OK
>revert
200 Reverted.
```

Will revert all changes made by this config script.

3.2.3 check

Listing 3: Check options

```
Syntax: check [explain]
>line add 111 type ivr
202 OK
>check
Will reload PBX users
```

Will check logical consequences of changed commands. It means if lines are unique or if pbx already exists etc. It reports if there is something wrong. If there is explain keyword, command will show verbose procedure, which acions should be taken to apply made changes. See listing 3.

3.2.4 get

Gets config or entire subtree. See listing 4.

Listing 4: Get options

```
Syntax: get [variable|context]
>get line 123
Line 123 is type ivr.
It is part of PBX main.
```

3.2.5 do

Listing 5: Do

```
Syntax: do {testcall|testprice|...}
>do testcall 123@main 234@main
Will call from main to main
Will redirect to 543 due to CFU set on 234
```

This command is used for actions which have no affect in database. Used mainly for debugging and testing. See listing 5.

Listing 11: Global userparms options

```
Syntax: userparm {add|remove|modify} 'name' \
          type {string|number|ip|uri} \
          [minlen len] [maxlen len] \
          [required] [{inherit|unique|pbxunique|serverunique}] [default 'defaultvalue']
Example:
userparm add location type string required
userparm add pin type number required unique minlen 4 maxlen 4
```

Listing 12: Global lineparms options

```
Syntax: lineparm {add|remove|modify} 'name' \
          type {string|number|ip|uri} \
          [minlen len] [maxlen len] \
          [required] [{inherit|unique|pbxunique|serverunique}] [default 'defaultvalue']
Example:
lineparm add skypename type string
lineparm add pin type number unique minlen 4 maxlen 4
```

3.3 System

Global parameters and config for entire system. It means for all servers. Here we are defining which servers are part of system and interconnecting them. See listing 6.

3.3.1 User

User management. See listing 7.

3.3.2 Line

Line management. See listing 8.

3.3.3 Phone

Phone management. See listing 9.

3.3.4 PriceList

Pricelist management. See listing 10.

3.3.5 UserParm

Global userparms management. See listing 11.

3.3.6 LineParm

Global userparms management. See listing 12.

Listing 6: System options

```
Syntax: {ca|key|server|user|number|phone}
Example:
ca crt 'certificationauthoritypemdata'
key add 'sshkey1' type ssh-rsa 'AAAA...'
key add 'sip.somewhere.net' type x509 'keydata'
server add sip.somwhere.net key 'sip.somewhere.net' ssh-key 'sshkey1'
server sip.somewhere.net
user add 'user1' givenname 'Pepa' sn 'Nos' \
        email 'email@ema.il'
number add 596398XXX
phone add 'gs-113' type gxp2000 \
        mac '00:01:02:03:04:05'
```

Listing 7: User options

```
Syntax: user {add|remove|modify} 'username' [userparms]
Example:
user add 'company1' type company brn 123456 name 'BestPractises, L.t.d.' \
         email 'best@com' emailpass
user modify 'company1' regno 123458
user add 'user1' type enduser givenname 'Pepa' surname 'Nos'
user modify 'user1' password 'secret'
```

Listing 8: Line options

```
Syntax: line {add|remove|modify} 'lineno' type 'type' protocol 'protocol' [lineparms]
Example:
line add '111222333' type ext protocol sip user 'user1' cfu '222333444'
```

Listing 9: Phone options

```
Syntax: phone {add|remove|modify} ['phonename'] type 'hwtype' [phoneparms]
Example:
phone add type gxp2000
phone add 'gxp2000-chef' type gxp2000 mac '00:01:02:03:04:05' serialno '123456' \
         user 'chef' line '444444444' pbx 'hq'
```

Listing 10: Pricelist options

```
Syntax: pricelist {add|remove|modify} type {operator|user} \
        direction {in|out} \
        subtype {call|notify|fax} \
        [from src] to dst \
        price currency start step
Example:
pricelist add type user direction out subtype call to 420. \
         price 1 currency CZK 1 1
pricelist add type operator direction out subtype call to 420. \
         price 0.5 currency CZK 30 30
pricelist add type operator direction out subtype call to 420. \
         price 0.5 currency CZK 30 30
```

4. **EXAMPLES**
5. **CONCLUSIONS**