FRAFOS ABC SBC Configuration Guide

Release 2.0.2

FRAFOS GmbH

December 12, 2013

Contents

This reference manual describes FRAFOS ABC SBC 2.0 GUI configuration.

Chapter 1

FRAFOS ABC SBC GUI

1.1 Login

To log into the FRAFOS ABC SBC GUI point your browser to external management IP address and login as an existing user.

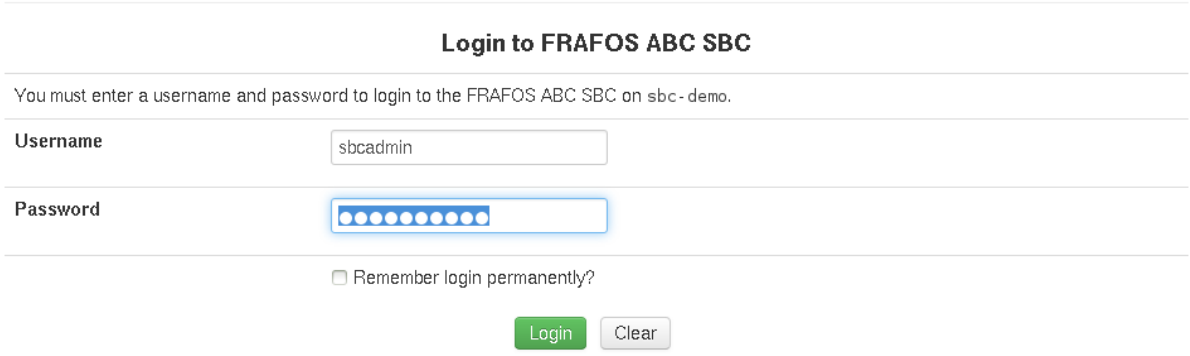


Figure 1.1: Login screen

The login from a browser can be stored permanently. If user wishes to do so he can use the checkbox *Remeber login permanently*.

By default following users exist:

- `sbcuser`, password `Sbc.User!`
- `sbcadmin`, password `Admin.SbC1?`

For more information about user accounts see `Installation Guide`.

1.2 Menu

The FRAFOS ABC SBC GUI menu is split into several categories:

Overview page gives summary overview about configured realms and call agents and inbound (A), and outbound (C) rules configured for them.

From the overview page user can jump directly to separate configuration screens and alter the configuration there.

Realms displays an administration screen for realms that allows to manage realms, call agents within these realms and inbound (A) and outbound (C) rules for the realms and their call agents.

Routing Allows to configure routing (B) rules, see *Routing rules* for more details.

Monitoring Servers for troubleshooting and monitoring purposes. Allows to investigate system and SBC related statistics, register cache content and list active calls. It is described in details in another document.

System Covers system related configuration (firewall, networking, SNMP), user management, FRAFOS ABC SBC license management, logical network interfaces (see *System configuration*).

HA High availability status overview and Virtual IP configuration.

Help Link to documentation in HTML format.

Logout Logs out logged user.

1.3 Applying configuration

After clicking Apply or Save button to save modifications the message *Warning: SBC configuration changed, activate to use* appears on the screen.

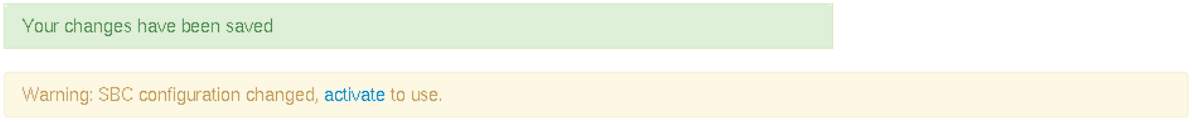


Figure 1.2: Activate changes

Click *activate* link if you want to apply the changes immediately and start processing calls with updated settings. However it is recommended to finish all needed configuration changes and then apply the whole configuration at once.

The configuration could also be activated by clicking the *Activate SBC configuration* link at the bottom of the *Overview page*.

Chapter 2

SBC Overview

This GUI page gives quick overview about SBC configuration and allows the user to jump directly to appropriate rule configuration for specific Realm or Call Agent.

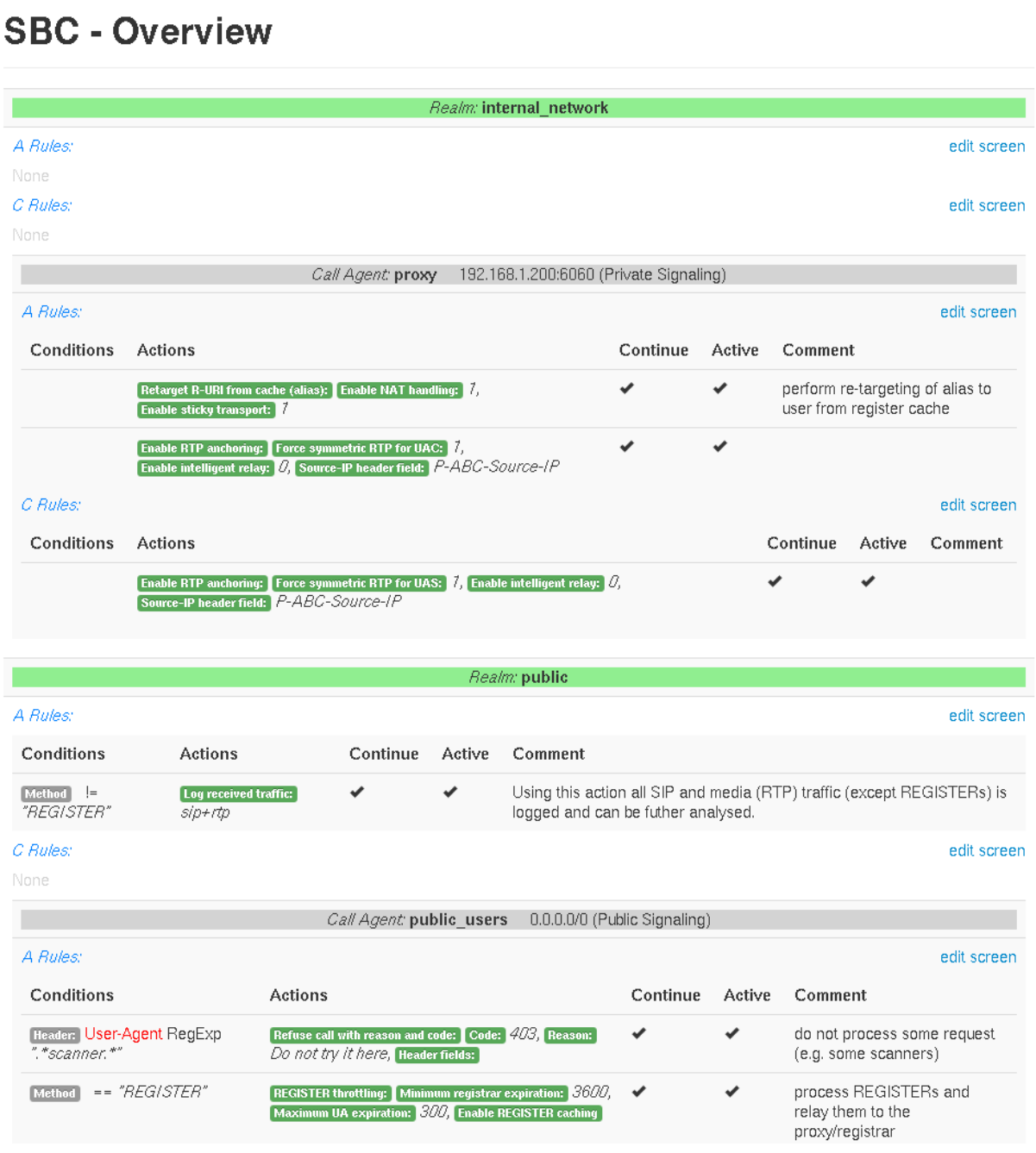


Figure 2.1: Overview screen

Chapter 3

Realms

This GUI page allows the user to manage FRAFOS ABC SBC realms, call agents in these realms and inbound (A) and outbound (C) rules for the realms.

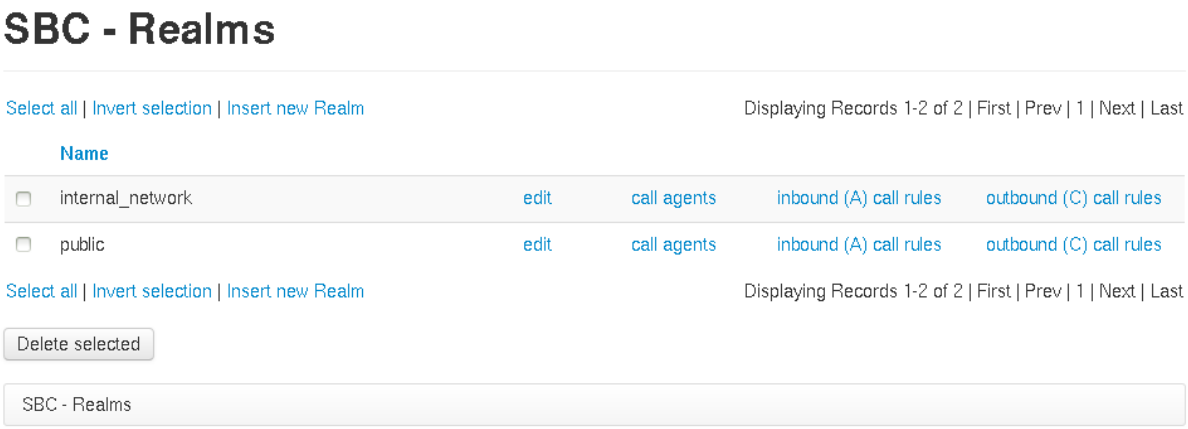


Figure 3.1: Realms

3.1 Create a new realm

Realms are identified by their name, there are no other options to be configured for them.

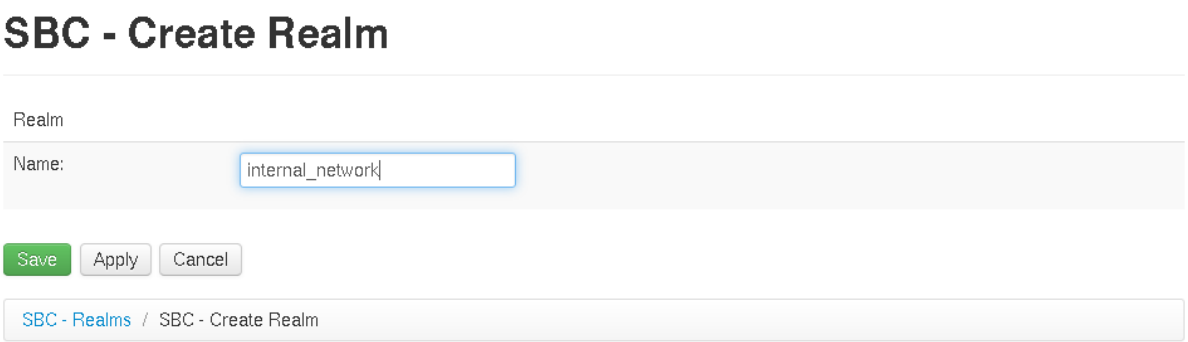


Figure 3.2: Create a new realm

3.2 Edit an existing realm

The realm name can be changed later on.

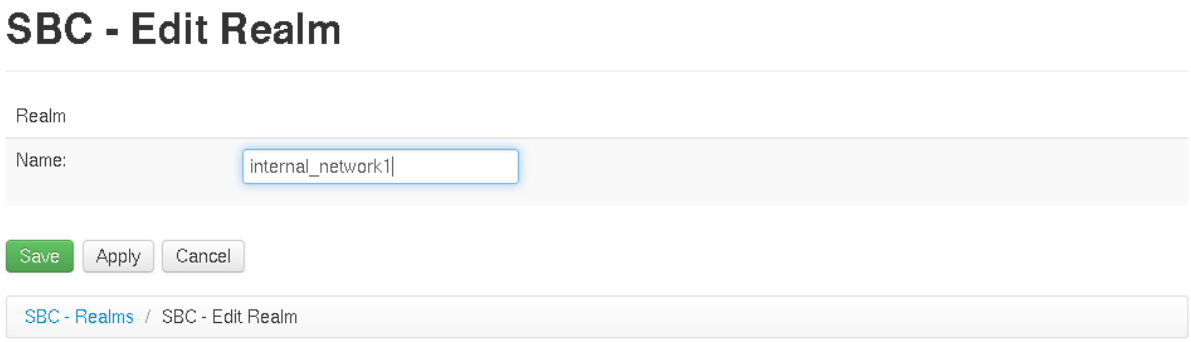


Figure 3.3: Edit an existing realm

3.3 Call Agents

After clicking to the link *call agents* the user can manage call agents for the realm and inbound (A) and outbound (C) rules for these call agents.

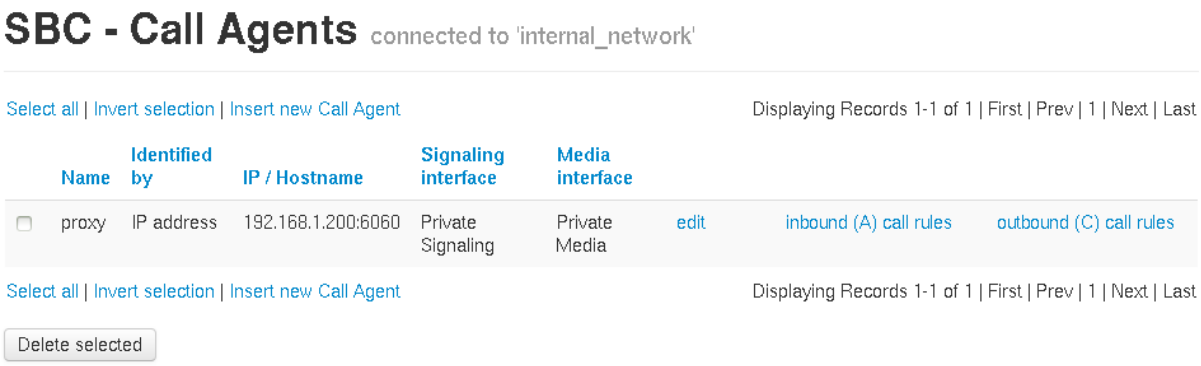


Figure 3.4: Call Agents

There are several options for each call agent to be configured:

Name identifies the call agent in GUI

Signaling interface determines the interface used for communication with this call agent. It is used for sending traffic to the call agent and for receiving the traffic as well (i.e. the call agent sending traffic to another interface than configured here is not matched).

Media interface determines the media interface used when relaying RTP to/from this call agent.

Identified by determines how the call agent is identified (for incoming traffic) or where the traffic is destined to (for outgoing traffic)

IP address determines call agent specified by single IP and port (optionally)

IP address range call agent specified by multiple IP addresses.

The value of 0.0.0.0/0 has special meaning and matches incoming traffic from everywhere if there is no better match with other call agents

DNS name call agent specified by a domain name

Call agent identified by DNS name can be used for outgoing traffic only!

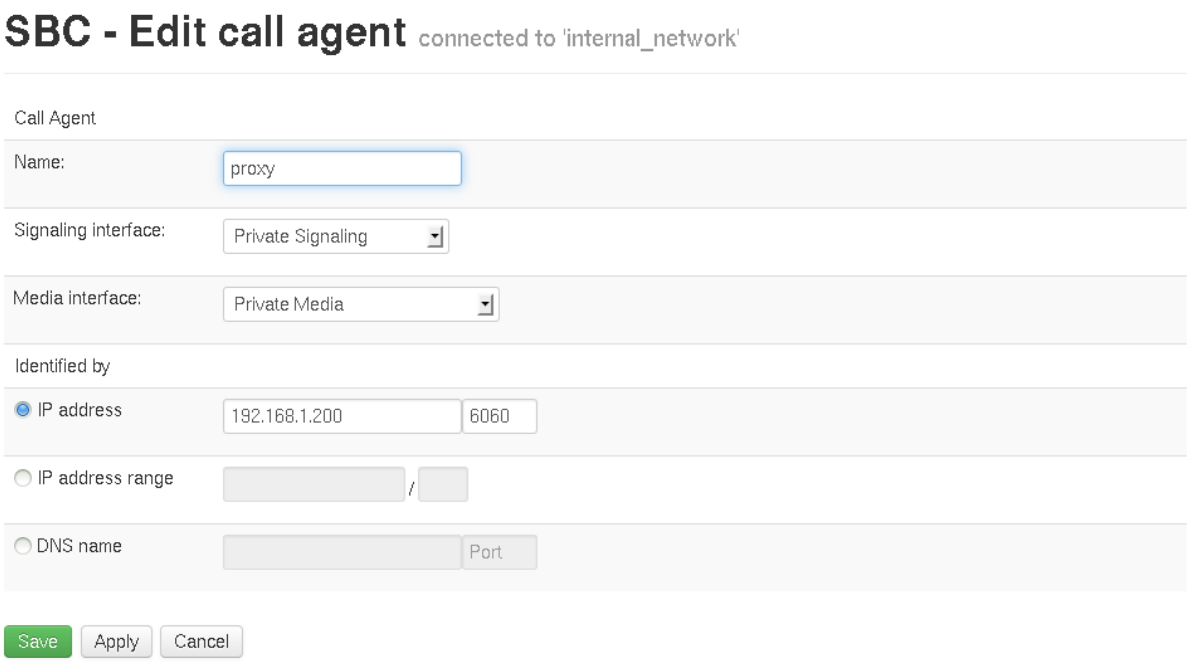


Figure 3.5: Call Agent parameters

3.4 Inbound (A) rules

Inbound rules can be configured for a realm or a call agent. These rules are applied on incoming traffic. Rules configured for a realm are applied for all call agents within the realm.

3.5 Outbound (C) rules

Outbound rules can be configured for a realm or a call agent. These rules are applied on outgoing traffic that is being routed to the realm or call agent. Rules configured for a realm are applied for all call agents within the realm.

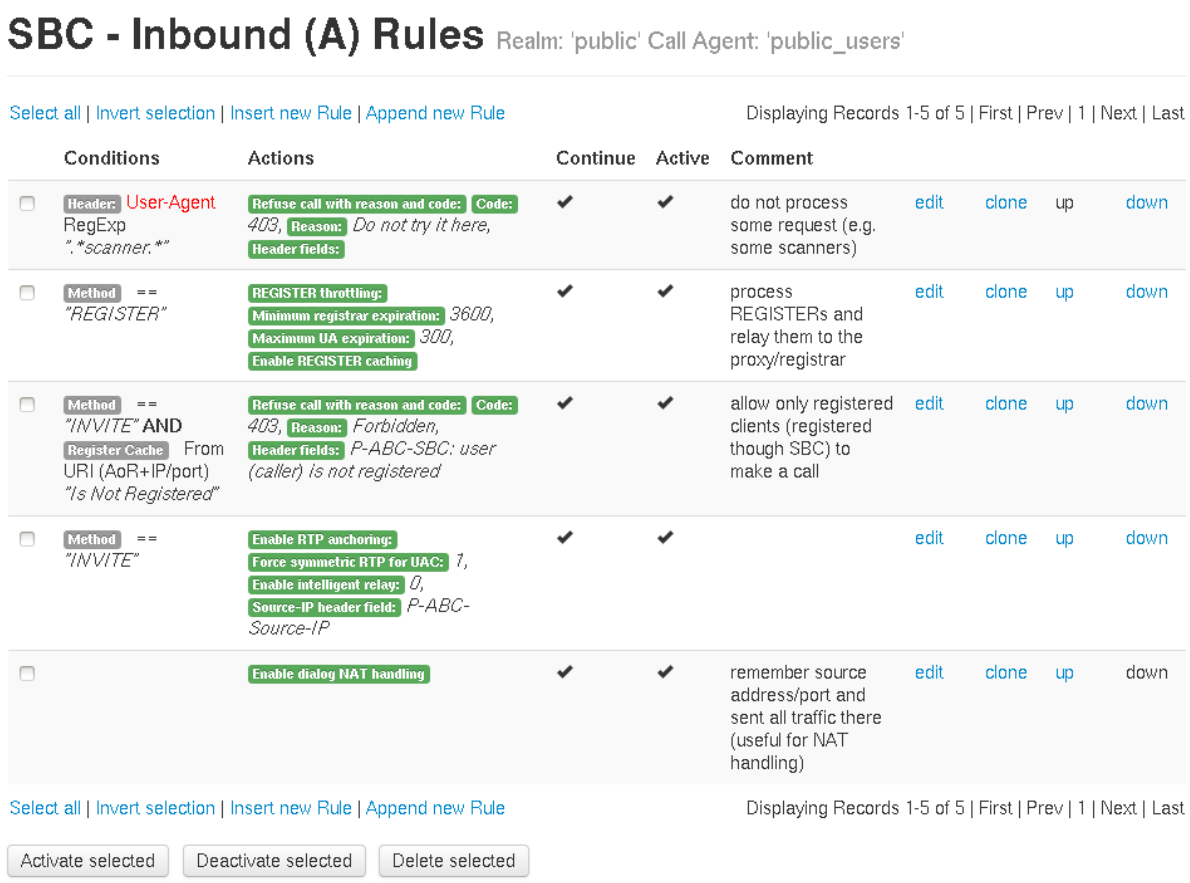


Figure 3.6: Inbound (A) rules

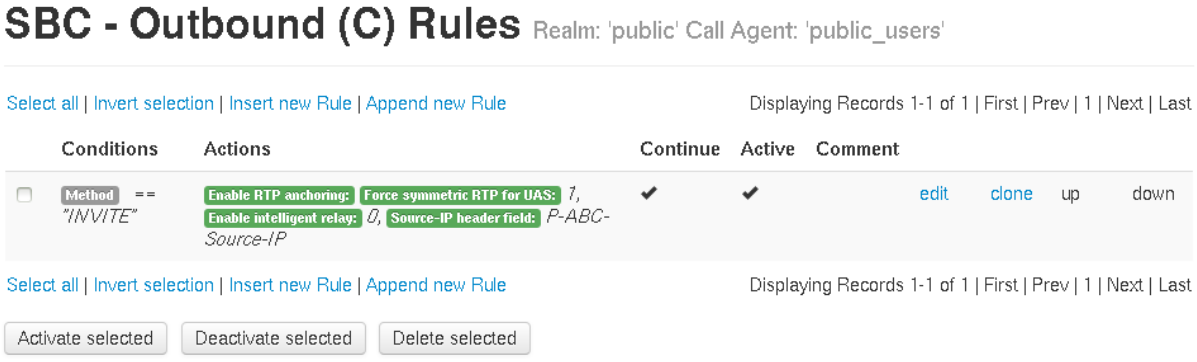


Figure 3.7: Outbound (C) rules

Chapter 4

Conditions

Conditions are used within inbound (A), routing (B) and outbound (C) rules to condition the actions or routing rules according to needs.

A condition consist of a condition type, operator and value. Possible operators are listed in tables *Operators supported within conditions* and *Operators supported within Register Cache condition*.

Table 4.1: Operators supported within conditions

Operator	Description
`==`	left operand equals to given value
`!=`	left operand does not equal given value
`RegExp`	left operand matches given regular expression
`does not match RegExp`	left operand does not match given regular expression
`begins with`	left operand starts with given string
`does not begin with`	left operand does not start with given string
`Contain`	right operand is contained in
`Contain RegExp`	sample described by right operand is contained in
`Do not contain`	right operand is not contained in

Table 4.2: Operators supported within Register Cache condition

Operator	Description
`From URI (AoR + Contact + IP/port)`	corresponding user is (not) registered
`From URI (AoR + IP/port)`	corresponding user is (not) registered
`Contact URI (Contact + IP/port)`	corresponding user is (not) registered
`To URI (AoR)`	corresponding user is (not) registered
`R-URI (Alias)`	corresponding user is (not) registered

The type of a condition defines the left operand for the operation. Please note that an expression in the right operand can contain *replacements*, but can not contain *backreferences*.

The table *Condition types* describes all available condition types. Operators `==`, `!=`, `RegExp`, `does not match RegExp`, `begins with`, `does not begin with` are supported unless specified otherwise.

Table 4.3: Condition types

Condition type	Description
Source Call Agent	Check the source call agent. Only operators == and != are supported.
Source IP	Check IP address the incoming request request was sent from.
Source port	Check port number the incoming request was sent from .
Inbound interface	Check local interface the incoming request was received on. Value has to be chosen from a list of configured signaling interfaces. Only operators == and != are supported.
R-URI	Check request URI
R-URI User	Check user part of request URI
R-URI User Parameter	Check parameter in username part of request URI (for example in R-URI like "sip:106;name=franta@domain.com" the parameter "name" can be checked for value "franta")
R-URI Domain	Check host part of request URI, can contain port number
R-URI URI Parameter	Check parameter of request URI.
From	Check From header field value.
From URI	Check value of From URI.
From User	Check user part of From URI.
From Domain	Check host part of From header URI, can contain port number.
To	Check To header field value.
To URI	Check value of To URI.
To User	Check user part of To URI.
To Domain	Check host part of To header URI, can contain port number.
Header	Check value of given SIP header.
Codecs	Check presence/absence of codecs within SDP. Right operand specifies codec name. Only operators `Contain`, `Contain RegExp` and `Do not contain` are supported.
Media Types	Check presence/absence of media type within SDP. Right operand specifies media type name (for example audio, video). Only operators `Contain`, `Contain RegExp` and `Do not contain` are supported.
Call Variable	Check call variable value using selected operator. Call variable name is given as condition parameter. The call variable has to be already defined by Set Call Variable action. Any condition referring to an undefined value returns FALSE as result.
Method	Check SIP request method. Value has to be chosen from a list of allowed methods. Only operators == and != are supported.
Register cache	Check content of register cache. Operators `From URI (AoR + Contact + IP/port)`, `From URI (AoR + IP/port)`, `Contact URI (Contact + IP/port)`, `To URI (AoR)`, `R-URI (Alias)` are supported

Chapter 5

Routing rules

Routing (B) rules are processing rules for finding out a destination for current request. If no routing rule exists for a request than such request can not be forwarded and is replied with 404 response.

Routing rule determines:

- destination call agent and realm so appropriate outbound (C) rules can be applied and interface configured for communication with selected call agent is used
- destination IP:port or domain name for routing the request
- possible R-URI modifications in outgoing request

A routing rule is applied when all its conditions are satisfied. In case a rule does not contain any conditions, the rule is always applied. Once a routing rule is applied no other routing rules are evaluated.

Three routing methods are available:

Set next hop Route calls to the specified address.

Set outbound proxy Add the specified address to a Route header and route calls accordingly.

Route via Request URI Route calls to the host address in the Request URI

If a call agent identified by IP address or domain name is selected, the outbound proxy address or next hop address is not mandatory. If it is not specified in the routing rule, it defaults to the address of the selected call agent.

Optionally the Request URI can be modified before sending the request out:

Update R-URI host indicates whether the host part of the R-URI of the outgoing INVITE request should be set to the address of the next hop or outbound proxy. Please note that the host value will be updated after applying both A and C call rules.

Replace R-URI host name through destination IP address causes to replace domain name in host part of the R-URI with destination IP address.

The “active” flag indicates whether the rule should be used during processing or not – inactive routing rules will be ignored.

5.1 DNS

If a domain name is choosen as the request destination (either destination call agent using a DNS name or routing via Request URI that contains domain name or the destination overriden in outbound proxy or next hop settings) the name is resolved before sending the request out.

Currently A and SRV DNS records are supported.

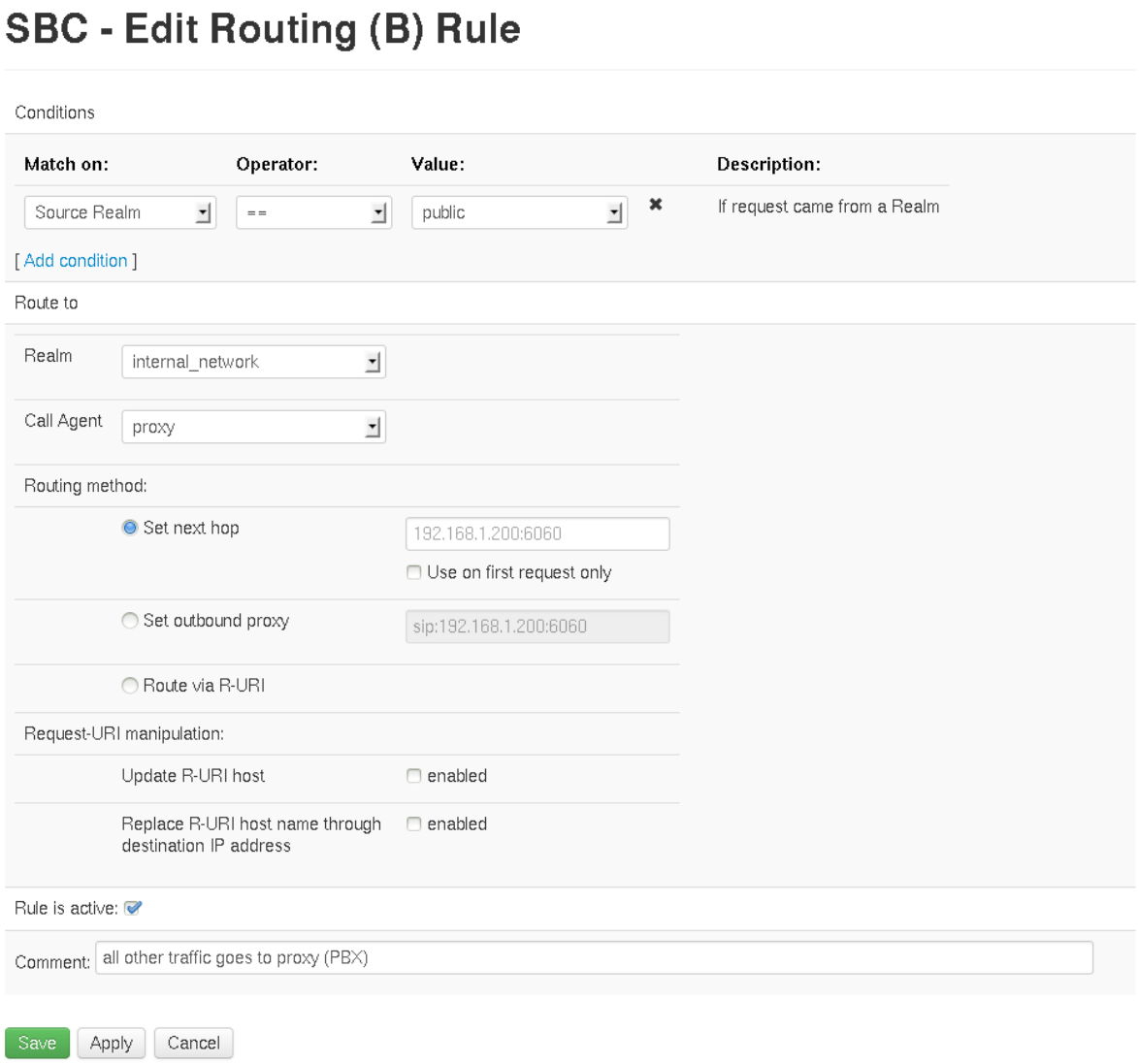


Figure 5.1: Routing Rule Example

5.1.1 SRV records

In case of SRV name resolving to several IP addresses FRAFOS ABC SBC does traffic load balancing according to destination parameters (priority and weight). Additionally failover between the multiple destination can be done.

Failover is applied for the dialog initiating request (initial INVITE) only. Once the dialog is established no more attempts to fail-over to next IP are done (sharing dialog state among different destinations can not be expected).

Failover is done in one of following cases:

- upon timer M timeout (internal timer not defined in RFC)

  Timer M is set by default to 8 seconds i.e. 1/4 of timer B/F (RFC transaction timer, default value 32s). If no reply is received within timer M, the next destination is tried. The call is failed once timer B/F hits and the timeout reply is propagated upstream.

- upon receiving 503 reply

  If the last destination replies with 503 (there are no more destinations to be tried) the 503 reply is propagated upstream.

Chapter 6

Actions

6.1 Enable RTP anchoring

Anchor FRAFOS ABC SBC into media path.

This action is necessary for successful NAT traversal or in cases FRAFOS ABC SBC should monitor RTP traffic between the endpoints. SDP body of all messages related to the call using this rule is altered to anchor RTP streams to the FRAFOS ABC SBC.

Parameters:

Force symmetric RTP to UAC/UAS If enabled, which is recommended, the media towards the SIP User Agent is sent to where the User Agent is sending its media from. This makes successful NAT traversal more likely for most of SIP devices. If this option is enabled in A-rules, it is applied towards caller (UAC). If enabled in C-rules, it is applied towards the called party (UAS).

Enable intelligent relay Allow media to flow directly between user agents behind the same NAT.

Source-IP header field Name of the header field used to save the source IP of the calling UA. This is necessary in order to determine whether or not the callee is behind the same NAT as the caller when the call passes the SBC twice.

6.2 Enable dialog NAT handling

Send all in-dialog messages using source IP/port of dialog-initiating request instead of SIP advertised IP/port.

This is safer in NATted environments than using IP addresses and port numbers advertised in the SIP protocol.

Parameters: none

6.3 Limit parallel calls

Limit number of parallel calls for a call agent or realm.

New calls exceeding this limit will be declined using the “403” SIP response.

If used in inbound (A) rules, the limitations refer to traffic coming from a Call Agent (if the action present in Call Agent’s rules) or realm (if present in realm’s rules). If used in outbound (C) rules, the limitations refer to traffic sent to a Call Agent or realm. For example:

> To limit the number of parallel calls from the FRAFOS ABC SBC to a realm, add a *Limit parallel calls* action to the realm’s outbound rules. To limit the number of parallel calls from a realm to the FRAFOS ABC SBC, add a *Limit parallel calls* action to the realm’s inbound rules.

Optionally the limits may be made more granular in that they don't refer to a whole Call Agent or realm but only to a partition of it. The partition is defined by the Key attribute: all dialogs with the same key are matched against a limit. For example

> if the key is "$si", the limit refers to all dialogs coming from the same IP address. The key can be arbitrarily defined using replacement expressions (see *Replacements*).

Parameters:

Limit parallel calls Number of allowed parallel calls.

Key attribute (optional) Limit is not applied to the Call Agent or Realm for that is the action configured but to the "area" identified by value of Key.

Is global key Specifies whether Key attribute value identifies global limit or is valid in context of Call Agent or Realm.

6.4 Limit CAPS

Set limit for call attempts per second for a call agent or realm.

If call rate exceeds this limit, new call attempts will be declined using the "403" SIP response.

The limit is applied similarly to *Limit parallel calls* depending whether the action is used in A or C rules to incoming or outgoing traffic and depending on Call Agent or Realm rules to Call Agent, Realm or another area if Key attribute is specified.

Parameters:

Limit CAPS Number of allowed call attempts per second.

Key attribute (optional) Limit is not applied to the Call Agent or Realm for that is the action configured but to the "area" identified by value of Key.

Is global key Specifies whether Key attribute value identifies global limit or is valid in context of Call Agent or Realm.

6.5 Limit Bandwidth (kbps)

Set limit for RTP traffic.

RTP packets exceeding this limit will be dropped.

If used in inbound (A) rules, the limitation refers to traffic coming from a Call Agent (if the action present in Call Agent's rules) or realm (if present in realm's rules). If used in outbound (C) rules, the limitation refers to traffic sent to a Call Agent or realm.

Parameters:

limit RTP traffic limit in kilobits per second.

Requires RTP anchoring activated by *Enable RTP anchoring* action.

6.6 Enable REGISTER caching

Cache contacts from REGISTER requests before forwarding and update contacts in forwarded REGISTER requests to point to FRAFOS ABC SBC to force all traffic for the registered client to be sent through FRAFOS ABC SBC.

REGISTER caching is necessary in environments where clients behind NAT register through SBC to a SIP registrar.

Apply this action in inbound (A) rules of a Call Agent or Realm representing clients that are behind a NAT.

Use *Retarget R-URI from cache (alias)* to restore and use the cached information for traffic from registrar.

Parameters: none

6.7 Retarget R-URI from cache (alias)

Rewrite Request URI with cached contacts.

Apply to messages sent to clients whose registration were cached previously using the *Enable REGISTER caching* action i.e. for example in inbound (A) rules of Call Agent standing for the registrar.

Parameters:

Enable NAT handling source IP and port of the REGISTER request are recovered and used instead of the ones from registered contact

Enable sticky transport use the same interface and transport over which the REGISTER was received

6.8 REGISTER throttling

Force SIP user-agents to shorten re-registration period while propagating the REGISTERs down to registrar at longer intervals.

This is useful to keep NAT bindings open without imposing the refreshing load on registrar.

Parameters:

Minimum registrar expiration expiration time in seconds used in direction to registrar (for example 600)

Maximum UA expiration maximum expiration time in seconds used in direction to User Agent Client (for example 30)

6.9 Save REGISTER contact in registrar

Act as local registrar - save contact from REGISTER request into internal database.

Use the action *Restore contact from registrar* to use the stored information.

Parameters: none

6.10 Restore contact from registrar

Use contact stored into internal registrar database using the action *Save REGISTER contact in registrar*.

Parameters: none

6.11 Increment SNMP counter

Increments given user-defined counter by given value.

The counter value can be queried via SNMP.

Parameters:

Counter name name of the counter to be incremented

Increment value value added to the counter

6.12 Log received traffic

Log messages sent and received within a call (dialog) in PCAP format. Every message log is stored as event and accessible through GUI *Monitoring → Events*.

Parameters:

Logged information describes what information is to be logged:

- SIP only

  Sent and received signaling traffic seen by SEMS process is saved into the log file. Generates "message-log" event pointing to the log file.

- SIP and RTP

  Sent and received signaling traffic and sent and received RTP traffic seen by SEMS process is saved into the log file. Generates "message-log" event pointing to the log file.

- SIP (real capture)

  Generates filter for filtering out signaling traffic from real traffic capture. Generates "pcap-filter" event that needs to be post-processed to generate the log file.

- SIP (real capture) and RTP

  Generates filter for filtering out signaling traffic from real traffic capture and merges the result with logged RTP traffic. Generates "pcap-filter" event that needs to be post-processed to generate the log file.

Warning: This action should not be used multiple times. Especially real capture can not be used at the same time as "SIP only" or "SIP and RTP". In such case error is reported in SEMS process log and only the first used logging action takes effect.

Important: Postprocessing of "pcap-filter" events can consume significant amount of resources (CPU, I/O) and take significant amount of time depending on the machine load and amount of data in the pcap files.

Important: Logged SIP and RTP traffic is not a real traffic capture and does not need to exactly match traffic captured on the network. It is a tool for catching application layer problems rather than network or transport layer.

Its key advantage is acceptably low resource consumption in comparison with real traffic captures.

This method of logging need not to be sufficient for all kind of problems, if so "SIP (real capture)" should be used (RTP is always logged, not captured!).

6.13 Set Call Variable

Define a variable with the specified name and value.

Once defined, value of the call variable can be tested in a *Call Variable* condition and/or referred to from any action using `$V(gui.varname)` replacement, where varname is the variable name (see *Replacements*).

Parameters:

Name identifies the variable to be set

Value contains the requested value.

Name and Value parameters may include replacement expressions (see *Replacements*).

6.14 Read call variables over REST

Read call variables from remote server using HTTP query.

This allows FRAFOS ABC SBC administrators to create fairly complex call processing logic outside the SBC in a web programming environment. The action passes needed information about the call attempt to a web server in form of a URI, and collects results from an HTTP answer. The results are used in further rule processing as Call variables.

Parameters:

URL resource identifier to be queried.

Result of the query is handled as list of call variables to be set. Expected format of data is text in format:

```
variable1=value1
variable2=value2
```

6.15 Append to RURI user

Append suffix to user part of Request URI.

The result is accumulated if the action is applied several times in the same rule or in different rules.

Parameters:

Suffix text to be added to the user part of R-URI. It may include replacement expressions.

6.16 Prefix RURI user

Add prefix to user part of Request URI.

The result is accumulated if the action is applied several times in the same rule or in different rules.

Parameters:

Prefix text to be prepended to the user part of R-URI. It may include replacement expressions.

6.17 Set RURI

Replace request URI with a new value.

Parameters

Value new value of the Request URI.

6.18 Set RURI host

Replace host(:port) part of Request URI with a new value.

Parameters:

Value new host (:port) part of the Request URI

6.19 Set RURI parameter

Add or replace parameter of Request URI.

Parameters:

Name the Request URI parameter name

Value new value of the Request URI parameter

6.20 Set RURI user

Replace user part of request URI with a new value.

Parameters:

Value new user part of Request URI

6.21 Strip RURI user

Strip the specified number of leading characters from user part of Request URI.

The result is accumulated if the action is applied several times in the same rule or in different rules.

Parameters:

Count number of charactes to be stripped

6.22 Set From

Set From header field value.

Parameters:

Value new From header field value in form “User name” <sip:user@domainname>

6.23 Set From display name

Set display name part of From header.

Parameters:

Value new display name value

6.24 Set From host

Replace host(:port) in From header field URI with a new value.

Parameters:

Value new host part of the From header

6.25 Set From user

Replace user part in From header field URI with a new value.

Parameters:

Value new value of From URI user part

6.26 Set To

Replace To header field with a new value.

Parameters:

Value new value of the To header field

6.27 Set To display name

Replace display name in To header field with a new value.

Parameters:

Value new value of the To header field display name

6.28 Set To host

Replace host part of URI in To header field with a new value.

Parameters:

Value new host part of the To header field

6.29 Set To user

Replace user part of URI in To header field with a new value.

Parameters:

Value new user part of the To header field

6.30 Enable SIP Session Timers (SST) callee leg

Enable session timer for callee (B) leg.

Parameters:

Session Expiration session expiration in seconds

Minimum Expiration minimum session expiration in seconds

6.31 Enable SIP Session Timers (SST) caller leg

Enable session timer for caller (A) leg

Parameters:

Session Expiration session expiration in seconds

Minimum Expiration minimum session expiration in seconds

6.32 Add Header

Append a header field to a SIP message.

The *Add header* action will add the header with the specified value even if the header is present in a *Set header blacklist* actions.

The parameters may include replacement expressions.

Parameters:

Header name name of the header field to be added

Header value value of the header field to be added

6.33 Remove Header

Exclude all occurences of a header field with the specified name. The action is applied to initial message as received, newly added header fields are not removed.

Parameters:

Header name name of the header field to be removed

6.34 Set header blacklist

Excludes all occurences of listed header fields. Blacklists are applied to the final appearance of the INVITE request after applying both A and C rules. They are applied one by one in the order in which they are defined in the rules.

Parameters:

Header list comma-separated list of header field names to be blacklisted

6.35 Set header whitelist

Removes all occurences of header fields that are neither listed nor mandatory in the SIP protocol. Whitelists are applied to the final appearance of the INVITE request after applying both A and C rules. They are applied one by one in the order in which they are defined in the rules.

Parameters:

Header list comma-separated list of header field names to be whitelisted

6.36 Translate reply code

Translate code and reason of received reply for the specified reply code to new code and reason.

Parameters:

Original code reply code to be translated (for example "606")

Replacement new reply code and SIP reason phrase (for example "480 Temporarily Unavailable")

6.37 Refuse call with reason and code

Refuse incoming call. If this action is executed, rules processing stops immediatelly.

Parameters:

Response code

Response phrase

Header fields (optional) additional header fields to be attached to the generated response. Replacement expressions can be used in response phrase and header field, multiple header fields can be added by putting rn between them.

6.38 Use transport

Enforce non-UDP transport for outgoing requests.

Only TCP is supported for now.

Parameters:

Transport transport to be used

6.39 Enable transparent dialog IDs

Use the same dialog ID (Call-ID, tags) on both sides of a B2B call (A leg and B leg). If this action is not enabled, the FRAFOS ABC SBC changes dialog ID.

Some see unchanged CallId as a security concern because it may contain caller's IP address. However transparent IDs make troubleshooting and correlation of call legs much easier.

Additionally, features like call transfers and authentication with some SIP implementations (Asterisk) may require the use of transparent dialog IDs.

Parameters: none

6.40 Set codec blacklist

Exclude all blacklisted codecs from SDP offer.

Parameters:

Codecs comma-separated list of codec names.

6.41 Set codec whitelist

Exclude all but listed codecs from SDP offer.

Parameters:

Codecs comma-separated list of codec names.

6.42 Set codec preferences

Allows to order codecs in SDP in each direction

Parameters:

A leg comma-separated list of codec names for A leg, codecs are ordered according to priority

B leg comma-separated list of codec names for B leg, codecs are ordered according to priority

6.43 Set media blacklist

Exclude all blacklisted media types from SDP.

Parameters:

Codecs comma-separated list of media types (audio, video, image, ...) to be blacklisted

6.44 Set media whitelist

Exclude all but listed media types from SDP.

Parameters:

Codecs comma-separated list of media types (audio, video, image, ...) to be allowed.

6.45 Activate transcoding

Allow transcoding for given codecs.

These codecs are introduced to SDP offer and if the other party accepts one of these, the media stream is transcoded.

For example when "pcmu,pcma" is configured, caller only offers PCMU only and the called party PCMA, SBC transcodes from PCMU at one side to PCMA at the other side and the other way around.

Parameters:

Codecs comma-separated list of codecs for which transcoding shall be activated

6.46 Drop early media

Do not forward incoming RTP within an early session (i.e. no media will be forwarded until the dialog is fully established).

Supported for audio only.

Parameters: none

Chapter 7

Replacements

Any parameter of any rule action can contain replacements – a special string, a dollar ("$") sign followed by an identifier. Each instance of a replacement is replaced by its value. Replacement values are initialized to the corresponding parts of the incoming INVITE request and then modified by actions.

For example, $aU is a replacement for the User part of the P-Asserted-Identity header; $th is a replacement for the host part of the To header. The action Set R-URI with the parameter set to *sip:$aU@$th* combines mentioned parts of P-Asserted-Identity and To headers of the incoming INVITE request and puts them into Request URI of the outgoing INVITE request.

All supported replacements are listed below in Value Replacement Reference.

Please note that when a replacement is used, special characters should be escaped as follows:

- \ → \\
- $ → \$

Table 7.1: Value Replacements Reference

Replacement group	Replacements	Description
$r	$r	R-URI
	$ru	R-URI URI
	$rU	R-URI User
	$rd	R-URI Domain (host:port)
	$rh	R-URI Host
	$rp	R-URI Port
	$rP	R-URI Parameters
$f	$f	From header
	$fu	From URI
	$fU	From User
	$fd	From Domain (host:port)
	$fh	From Host
	$fp	From Port
	$fn	From Display name
	$fP	From Parameters
	$ft	From Tag
	$fH	From Headers
$t	$t	To header
	$tu	To URI
	$tU	To User
	$td	To Domain (host:port)
	$th	To Host
	$tp	To Port
	$tn	To Display name

Continued on next page

Table 7.1 – continued from previous page

	$tP	To Parameters
	$tt	To Tag
	$tH	To Headers
$a	$a	P-Asserted-Identity header
	$au	P-Asserted-Identity URI
	$aU	P-Asserted-Identity User
	$ad	P-Asserted-Identity Domain (host:port)
	$ah	P-Asserted-Identity Host
	$ap	P-Asserted-Identity Port
	$aP	P-Asserted-Identity Parameters
	$at	P-Asserted-Identity Tag
	$aH	P-Asserted-Identity Headers
$p	$p	P-Preferred-Identity header
	$pu	P-Preferred-Identity URI
	$pU	P-Preferred-Identity User
	$pd	P-Preferred-Identity Domain (host:port)
	$ph	P-Preferred-Identity Host
	$pp	P-Preferred-Identity Port
	$pP	P-Preferred-Identity Parameters
	$pt	P-Preferred-Identity Tag
	$pH	P-Preferred-Identity Headers
$c	$ci	Call-ID
$s	$si	Source (remote) IP address
	$sp	Source (remote) port number
$R	$Ri	Destination (local/received) IP address
	$Rp	Destination (local/received) port number
	$Rf	local/received interface id (0=default)
	$Rn	local/received interface name
	$RI	local/received interface public IP
$H	$H(headername)	value of header headername (Note: not all headers are available here)
	$HU(headername)	header headername (as URI) User
	$Hd(headername)	header headername (as URI) domain (host:port)
$V	$V(gui.varname)	value of call variable varname
$B	$B(cnum,rnum)	value of backreference with *rnum* number from the condition with *cnum* number
$_	$_u(value)	value to uppercase
	$_l(value)	value to lowercase
	$_s(value)	length of value
	$_5(value)	MD5 of value
	$_r(value)	random number 0..value, e.g. $_r(5) gives 0, 1, 2, 3 or 4

Chapter 8

Regexp backreferences

Rule conditions allow to match strings against regular expression (RegExp). Backreferences allow actions to refer to a substring of strings matched previously in conditions. Backreferences are identified by two indeces: number of condition in a rule and number of a substring inside the regular expression. Both numbers begin from 1, so $B(1.1) stands for the first substring in the fist RegExp condition.

To extract a substring, you must identify the part of Regular expression exactly matching it and enclose it in parenthesis. For example, if you would like to refer to the numerical substring enclosed in alphabetical characters like “aaa111bbb”, you could define the regular expression as “([a-z]*)([0-9]*)[a-z]*)”. The backreference is then $B{1.2): the second substring of the first condition.

Chapter 9

Common use cases

9.1 NAT traversal

For successful NAT traversal it is needed to configure FRAFOS ABC SBC for RTP relaying using the action *Enable RTP anchoring*. In most cases it is required to set *Force symmetric RTP* for both endpoints: in inbound (A) rules for caller and in outbound (C) rules for callee.

Additionally the action *Enable dialog NAT handling* should be used to send in-dialog messages to source IP:port of the dialog initiating requests instead of to the IP:port advertised in SIP headers.

Common way of handling REGISTER requests in NATted environment is:

- use register cache for rewriting contacts so traffic to such registered client is sent through FRAFOS ABC SBC (necessary for successful NAT traversal)
- activate REGISTER throttling to force clients to keep NAT bindings open regardless they support this feature explicitly or not

For such case there should be inbound (A) rule for REGISTER request from a client to activate register caching (see figure *Activate register caching*) and another inbound (A) rule for using register cache content for requests from registrar (see figure *Use cached information*).

9.2 Traffic shaping

FRAFOS ABC SBC allows to impose limitations on traffic, both SIP signaling and RTP media.

Limits can be used to split available resources (for example the global limit on parallel calls given by license) to specific areas or as a part of DoS protection.

To configure limits following actions can be used:

- *Limit parallel calls*
- *Limit CAPS*
- *Limit Bandwidth (kbps)*

If such action is used in inbound (A) rules, the limitations refer to traffic coming from a Call Agent or realm. If used in outbound (C) rules, the limitations refer to traffic sent to a Call Agent or realm.

Optionally the limits may be made more granular in that they don't refer to a whole Call Agent or realm but only to a partition of it. The partition is defined by key: all dialogs with the same key are matched against a limit. For example, if the key is "$si", the limit refers to all dialogs coming from the same IP address. The key can be arbitrarily defined using replacement expressions.

If limit of parallel calls or call atempts per second is reached new calls trying to get over this limit are refused with 403 response.

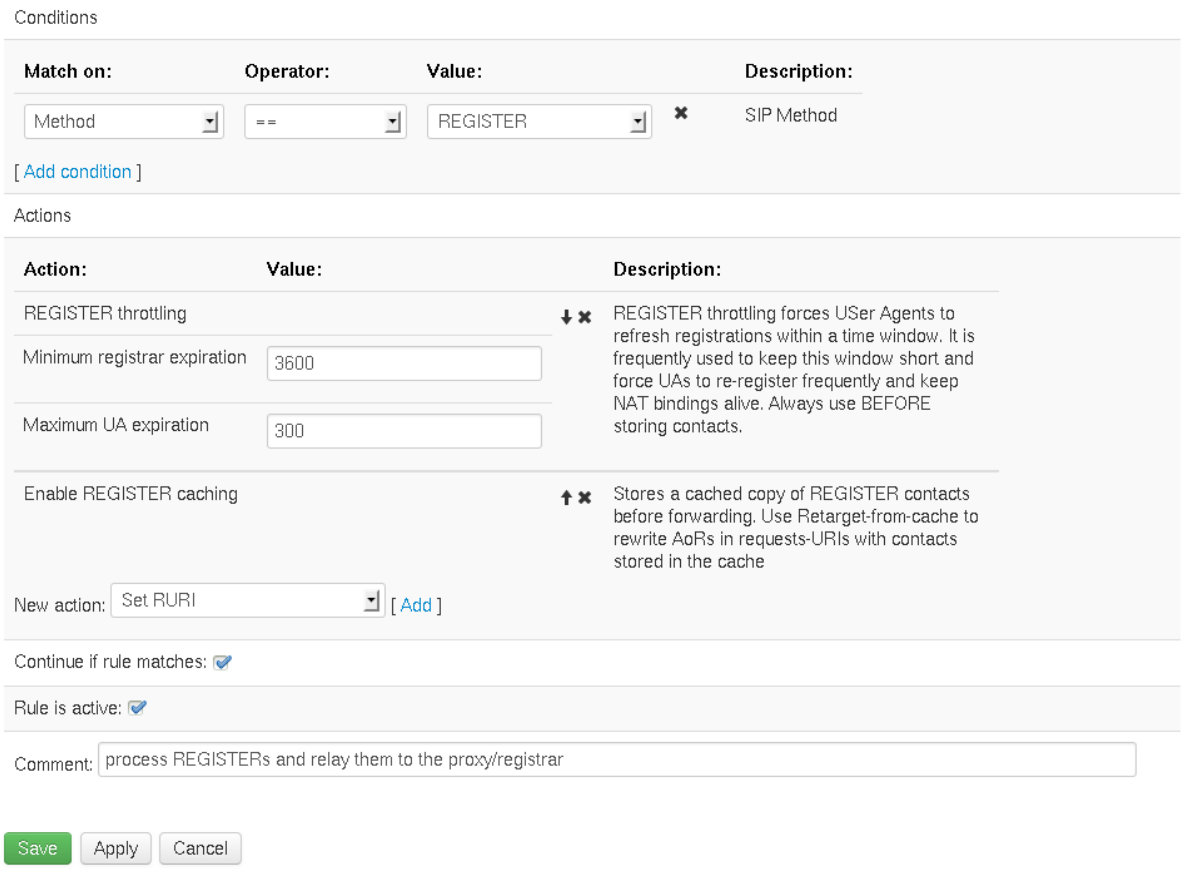


Figure 9.1: Activate register caching

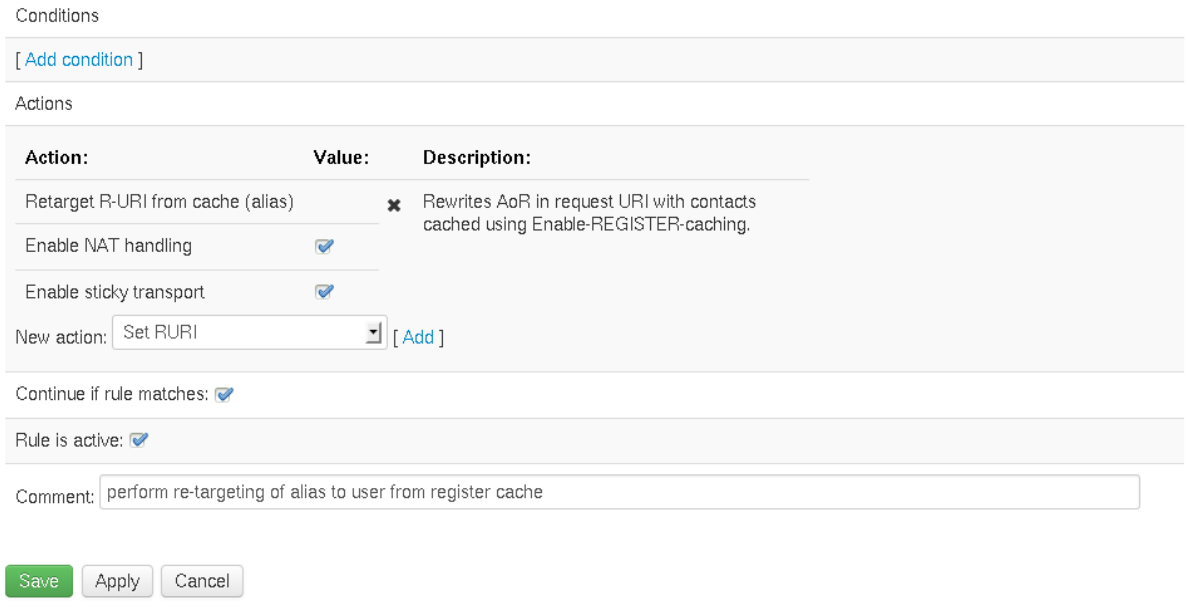


Figure 9.2: Use cached information

If the badwidth limit is reached RTP packets over this limit are dropped.

Additionally the operator can apply rules that will force codecs used for media communication using *Set codec blacklist*, *Set codec whitelist* or *Set codec preferences* to choose ones with lower bandwidth consumption or completely disable some media types using *Set media blacklist* or *Set media whitelist* actions.

9.3 Internal registrar

FRAFOS ABC SBC is able to act as a SIP registrar with limited capabilities. Though this is neither common nor recommended it may be suitable for demonstrations or very small "all-in-one" solutions.

Use following actions to handle REGISTER messages as a registrar:

- *Save REGISTER contact in registrar*
- *Restore contact from registrar*

Authentication is not supported by internal registrar explicitly (there is no internal user credentials database) but can be done using another actions, for example *Read call variables over REST*.

9.4 Mediation actions

FRAFOS ABC SBC offers wide range of actions that can be used to modify SIP messages including SDP bodies for best interworking between various SIP devices or for security purposes (topology hiding, remove internal SIP headers, limit codec usage).

- R-URI modifications
  - *Append to RURI user*
  - *Prefix RURI user*
  - *Set RURI*
  - *Set RURI host*
  - *Set RURI parameter*
  - *Set RURI user*
  - *Strip RURI user*
- To/From header modifications
  - *Set From*
  - *Set From display name*
  - *Set From host*
  - *Set From user*
  - *Set To*
  - *Set To display name*
  - *Set To host*
  - *Set To user*
- actions for generic SIP header manipulation
  - *Add Header*
  - *Remove Header*
  - *Set header blacklist*

  - *Set header whitelist*
- actions for altering replies
  - *Translate reply code*
- other signaling mediation actions
  - *Use transport*
  - *Enable transparent dialog IDs*
- media related mediation actions
  - *Set codec blacklist*
  - *Set codec whitelist*
  - *Set codec preferences*
  - *Set media blacklist*
  - *Set media whitelist*
  - *Activate transcoding*
  - *Drop early media*

9.5 Topology hiding

Topology hiding is appling *Mediation actions* to reduce visibility of one network internals to another network(s). The rules can be applied in inbound (A) or outbound (B) rules for realms representing the networks.

Because FRAFOS ABC SBC acts as a B2B user agent the most important part of the sensitive information (Via headers, Contacts, Routes) is hidden by nature.

Dialog ID (Call-ID, To-tag, From-tag) is by default hidden as well but it can be "unhidden" by applying the action *Enable transparent dialog IDs*. If applied, FRAFOS ABC SBC stops hiding dialog ID information for affected call to support scenarios that need this information propagated through SBC to the endpoints. For example features like call transfers or BLF may require transparent dialog IDs.

For hiding other information than specified above the user can apply:

Enable RTP anchoring to hide media connection related information. If applied, FRAFOS ABC SBC places itself into the media path and the endpoints see just the SBC addresses used for media traffic.

Set codec blacklist **or** ***Set codec whitelist*** to hide some of available media codecs to the other party

Set media blacklist **or** ***Set media whitelist*** to disable some of available media types (like video)

Set header blacklist **or** ***Set header whitelist*** to avoid forwarding of specific headers in all messages within affected dialog (call)

Remove Header to remove headers from initiating dialog request only (initial INVITE)

Using regular expressions together with *Add Header* and *Remove Header* actions it is possible to specify more sophisticated rules like changing specific parts of a header. For example replacing domain with "example.com" in a header field is shown in figure *Topology Hiding: overwriting domain*.

9.6 DoS protection

Security reasons are one of the key reasons for having an SBC. FRAFOS ABC SBC offers wide range of actions to fulfil customers needs.

One important part of security actions is based on traffic limits described above (see *Traffic shaping*).

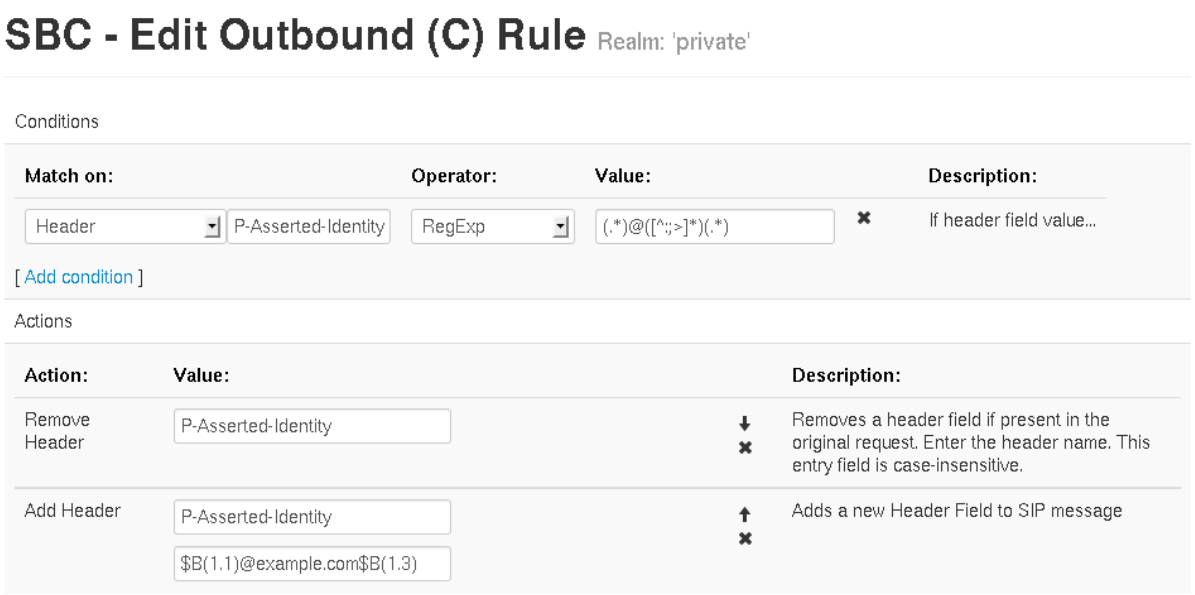


Figure 9.3: Topology Hiding: overwriting domain

Another actions that can be used to prevent different kind of attacks (not only DoS) are described in sections below.

Finally the administrator can tune firewall settings to properly match the attack conditions and filter out unwanted traffic at IP level.

9.6.1 Blocking a User Agent

There are well-known user agents like “friendly-scanner” that can be detected in inbound (A) rules according to User-Agent header or another parts of the SIP message. Such requests can be refused with *Refuse call with reason and code* action. (see figure *DoS protection: refuse calls from user agent*).

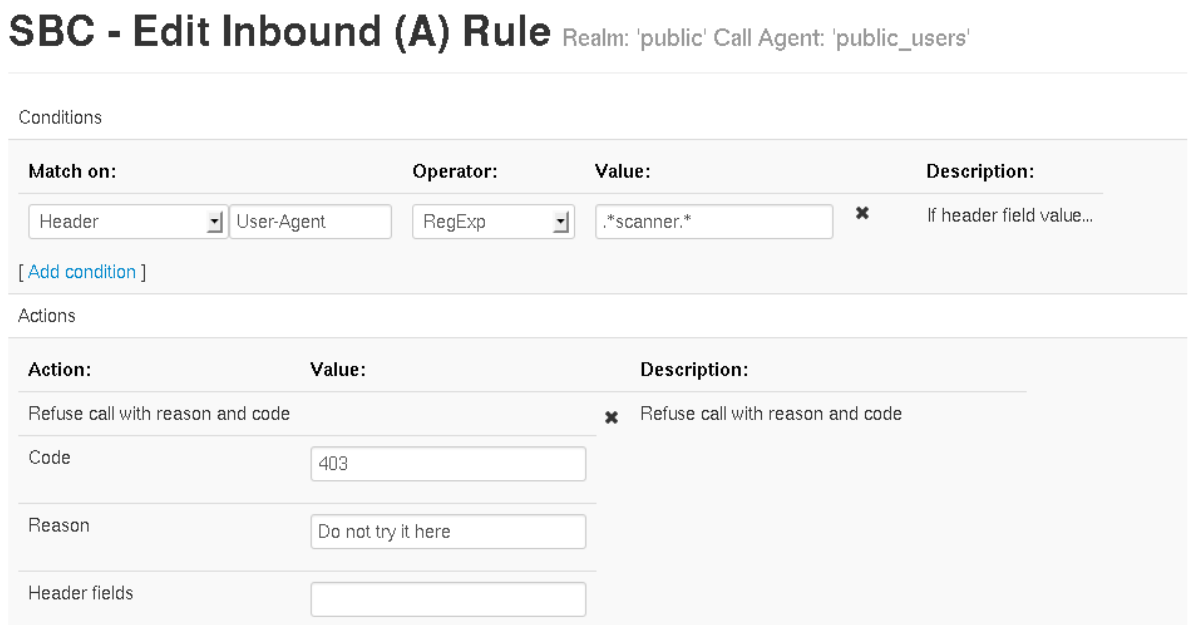


Figure 9.4: DoS protection: refuse calls from user agent

9.6.2 Blocking IP address

It is possible to block single IP address or multiple IP addresses matching a text pattern with actions configured with Source IP condition (see figure *DoS protection: refuse calls from an IP*).

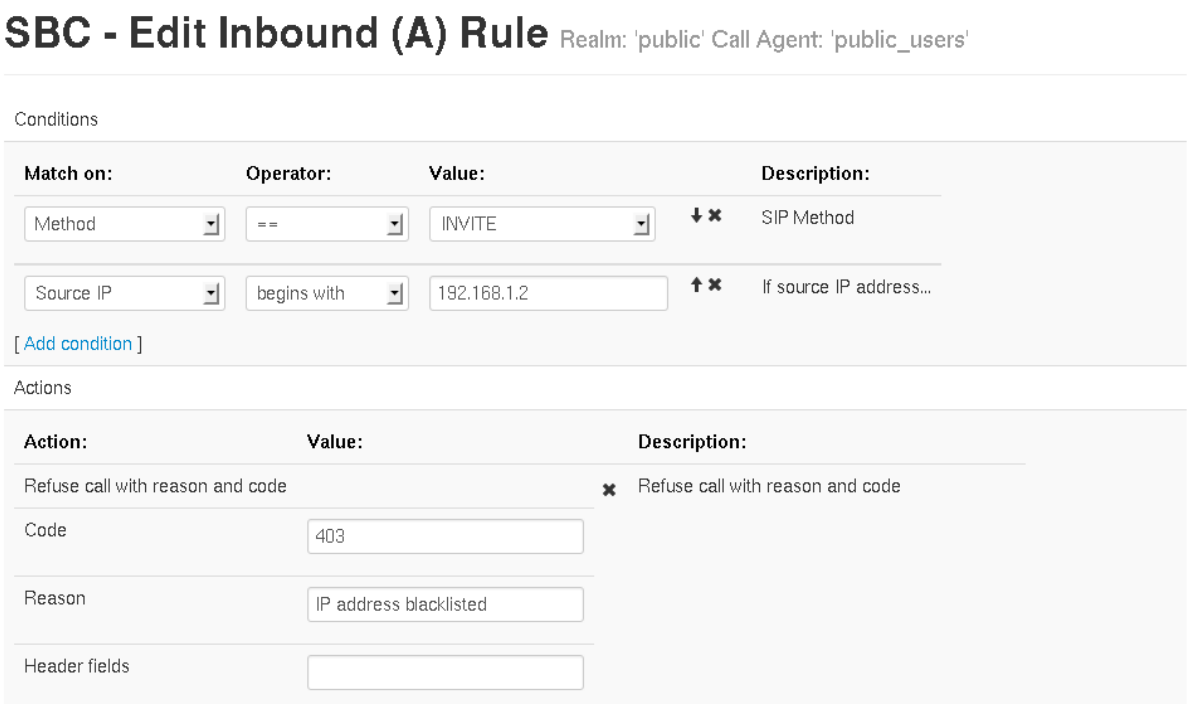


Figure 9.5: DoS protection: refuse calls from an IP

9.6.3 Blocking IP address range

The simpliest way to block a range of IP addresses is to create a call agent for such IP address range (*DoS protection: refuse calls from IP range (1)*) and create an inbound (A) rule for this call agent without conditions that will refuse all messages from it (*DoS protection: refuse calls from IP range (2)*) .

Figure 9.6: DoS protection: refuse calls from IP range (1)

9.7 Call Admission Control

Regulating traffic volume is described in details in *Traffic shaping*. Additionally the user can regulate incoming traffic using practices outlined in *DoS protection*.

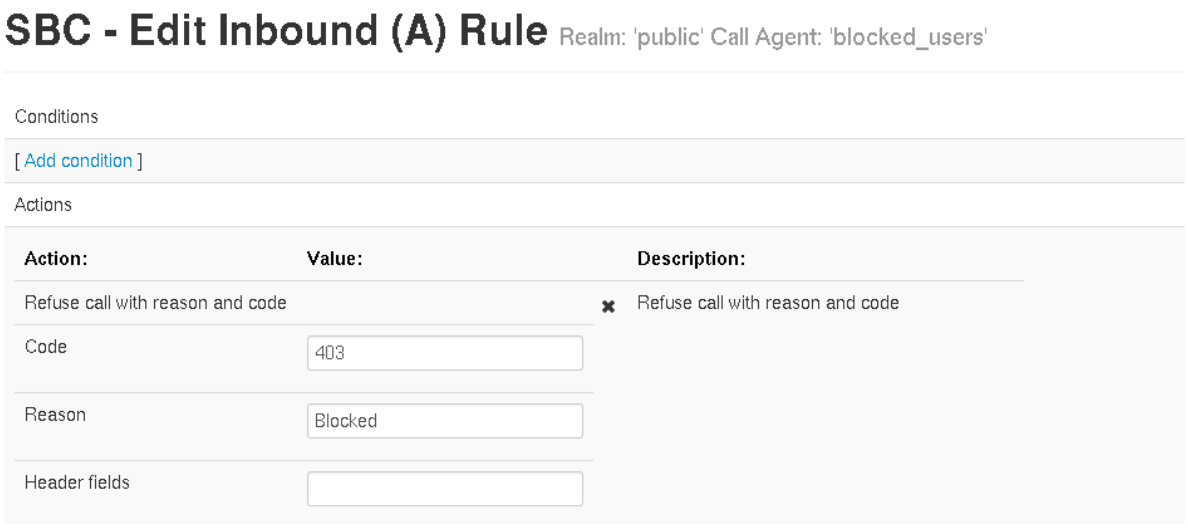


Figure 9.7: DoS protection: refuse calls from IP range (2)

Chapter 10

Advanced rule examples

10.1 Regexp backreferences

In this example (Figure 4) we save the scheme of the Request URI of the incoming INVITE request in a call variable called scheme and enforce the “sip” scheme for the R-URI of the outgoing INVITE request. Here the action uses two backreferences referring to the regular expression from the second condition. The first backreference refers to the scheme of the R-URI. The second backreference refers to request URI.

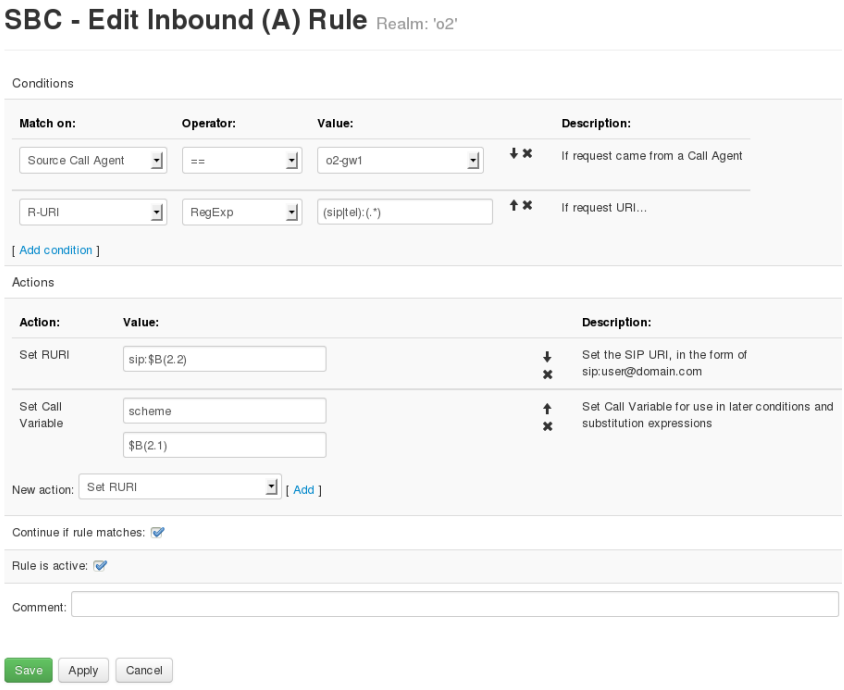


Figure 10.1: Regexp backreferences example

10.2 Call variables

In this example (Figure 5) we refuse the call when the Request URI is a “tel” URI. First we check the type of reqeust URI. If it has the value “tel”, we define the call variable refuse and refuse the call. We also put the value of the call variable scheme into the Reason header using a replacement pattern.

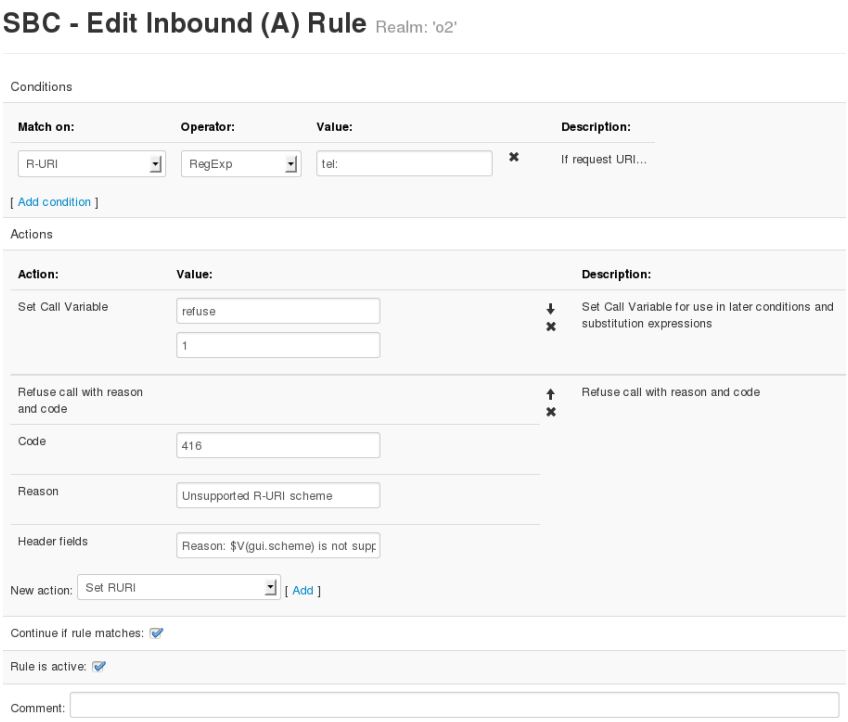


Figure 10.2: Call variables example

10.3 Replacements

In this example (Figure 6) we set up outgoing INVITE request as follows: set Request URI of the outgoing INVITE request to the user part of the P-Asserted-Identity header ($aU) combined with the host part of the To header ($th) of the incoming INVITE request; set host part of the To header to the value of the P-NextHop-IP header ($H(P-NextHop-IP))of the incoming INVITE request (the user part will not be changed); convert the user part and the host part of the From header into lower case (<sip:$_I($fU)@_I($fh)>).

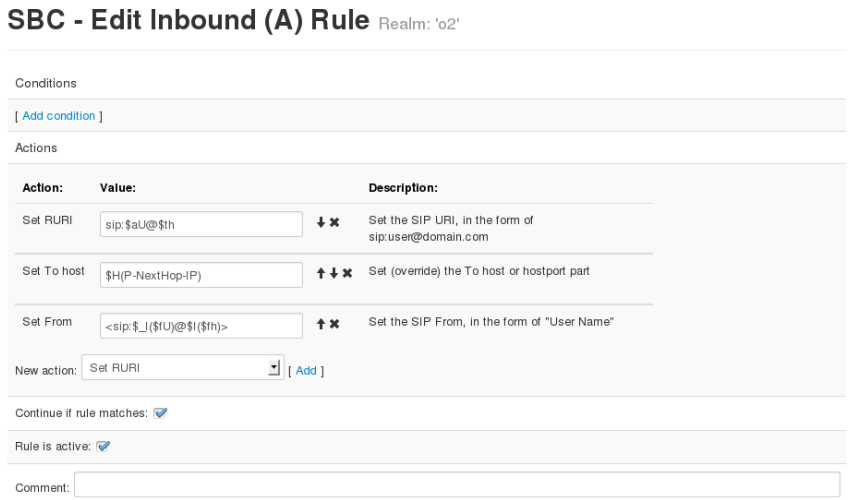


Figure 10.3: Replacement Rule Example

How the transformation rules modify the incoming INVITE is shown in the following figure *Diagram of transformed SIP message*.

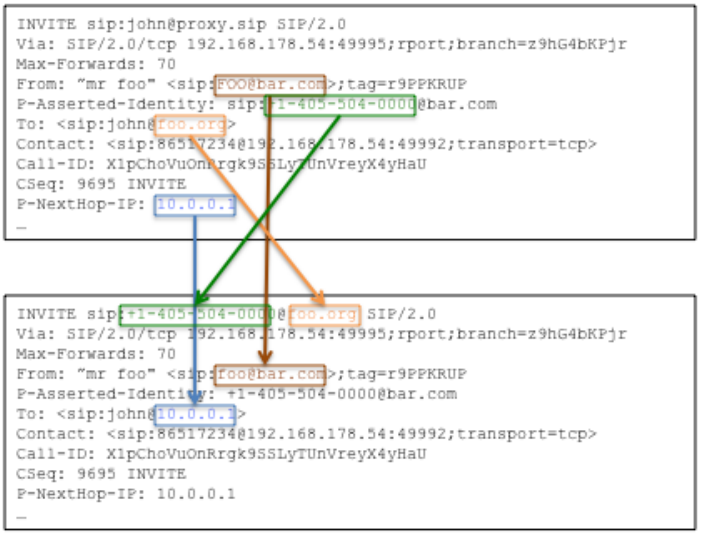


Figure 10.4: Diagram of transformed SIP message

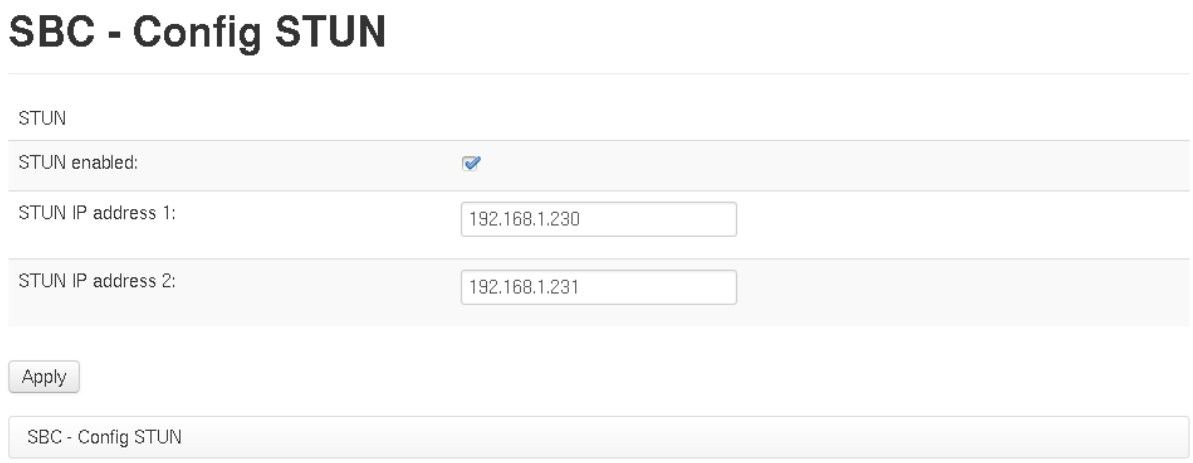
Chapter 11

System configuration

System configuration is available under menu *System*. This menu has several submenus described in their own sections below.

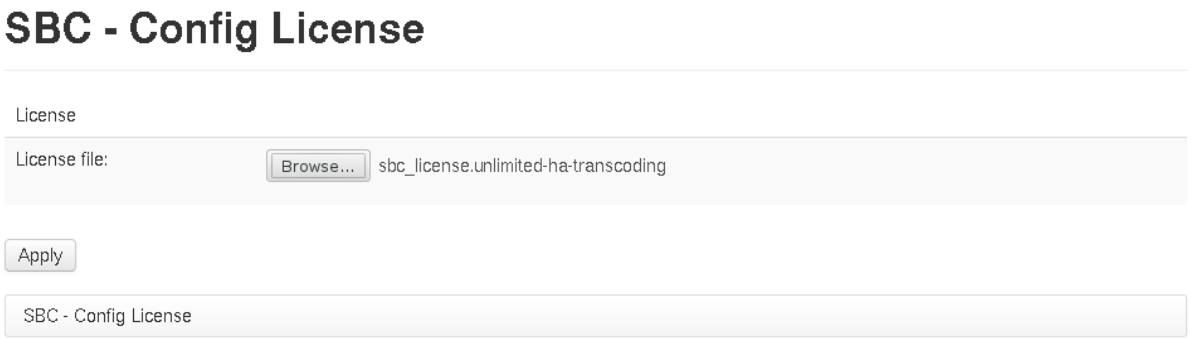
11.1 STUN

Allows to enable/disable STUN server and configure IP addresses of the STUN server.



11.2 License

Allows to display currently active FRAFOS ABC SBC license and upload a new one.

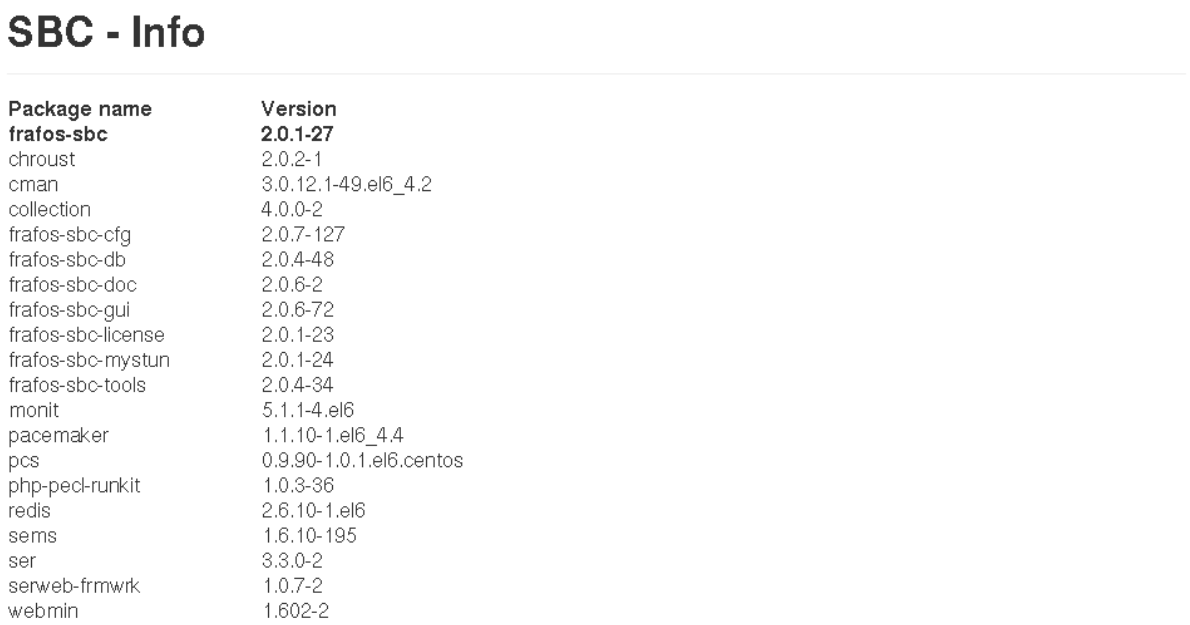


11.3 Users

Allows to add new, delete or update existing users. User management is in detail described in installation guide.

11.4 Info

Displays current FRAFOS ABC SBC package versions.



11.5 Firewall

Allows to configure firewall rules through GUI using standard webmin module. Firewall rules configuration are out of scope of this document.

11.6 Networking

Allows to configure network interfaces, routing, gateways, DNS and hostname addresses through GUI using standard webmin module. Detailed description is out of scope of this document.

11.7 Interfaces

This GUI page allows to manage logical interfaces above existing system interfaces. These logical interfaces serve as abstraction from system interfaces and are used in call agent configuration.

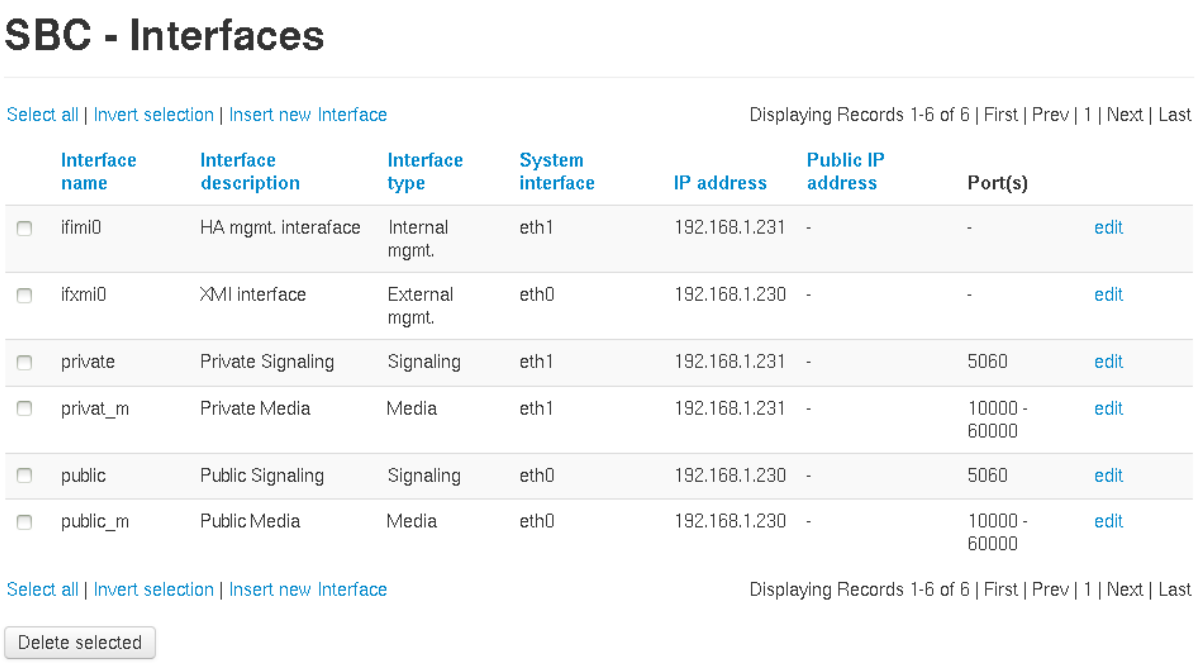
There are several types of logical interfaces used in FRAFOS ABC SBC:

Signaling Interface can be used for SIP traffic.

Media Interface can be used for media traffic.

External mgmt. Management interface for external communication (GUI, SNMP, SSH, ...).

Internal mgmt. Management interface for internal communication (HA).



When adding a new or modifying existing interface, the user has to choose following interface paramaters:

Interface name Identifies the interface. This name has to be equal on HA paired machines.

Interface type Specifies the interface type, can be one of values described above.

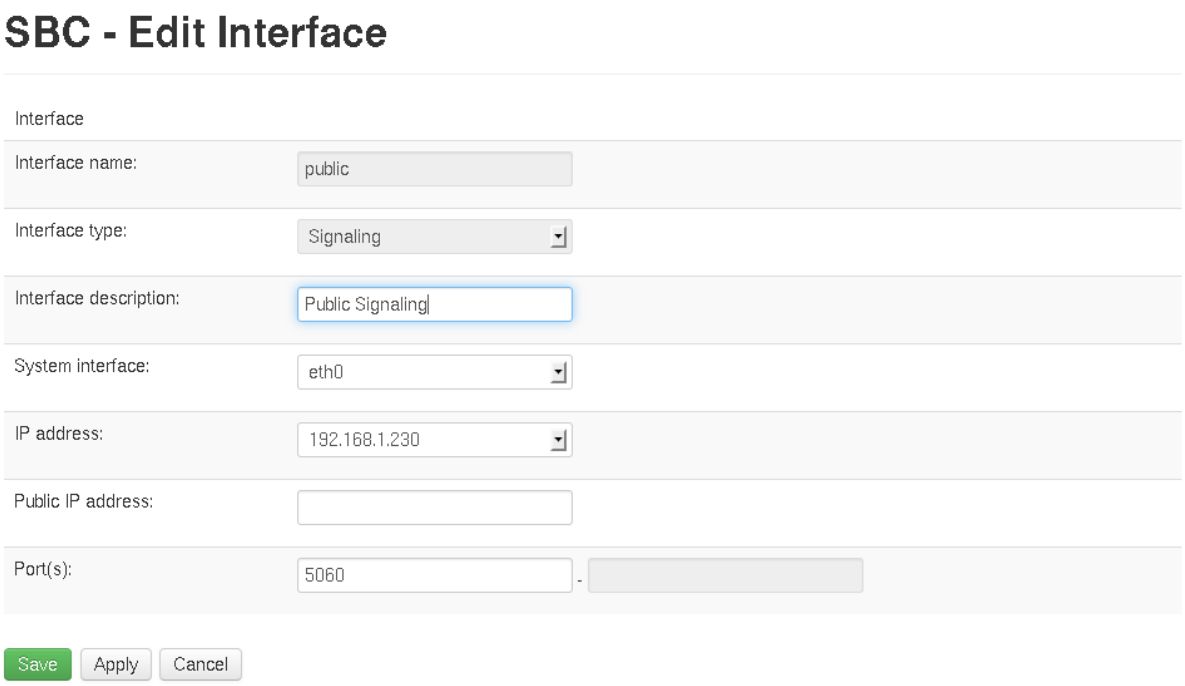
Interface description Describes the interface and is shown in other GUI pages like call agent configuration.

System interface One of existing system interfaces.

IP address One of the IP addresses assigned to the underlying system interface (including configured Virtual IPs).

Public IP address Address used in SIP messages or in SDP (in case of Media interface) if SBC is operating behind NAT that translates this Public IP address to the non-public one.

Port(s) Required for Media interfaces to specify port range usable for RTP traffic or single port in case of Signaling interface.

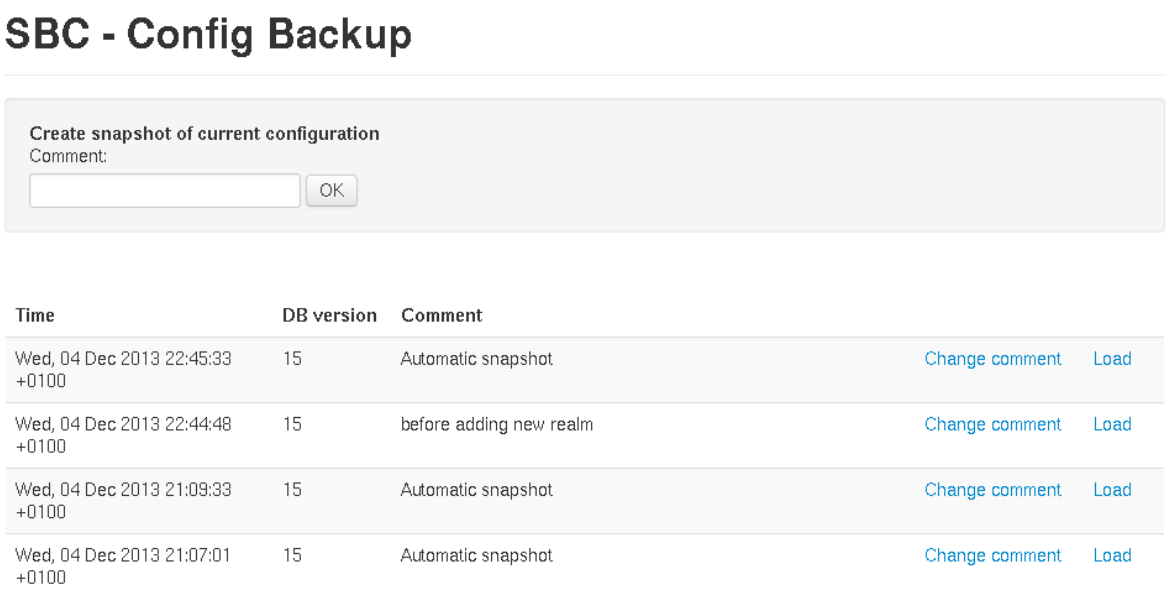


More information about interfaces can be found in `Installation Guide`.

11.8 Old configurations

Allows the administrator to save current FRAFOS ABC SBC configuration or return to a previously stored configuration version.

Note: The configuration snapshots are created automatically when activating changes.



11.8.1 Saving configuration

To explicitly save a configuration fill the *Comment* and click *OK* button on the GUI page.

11.8.2 Restoring configuration

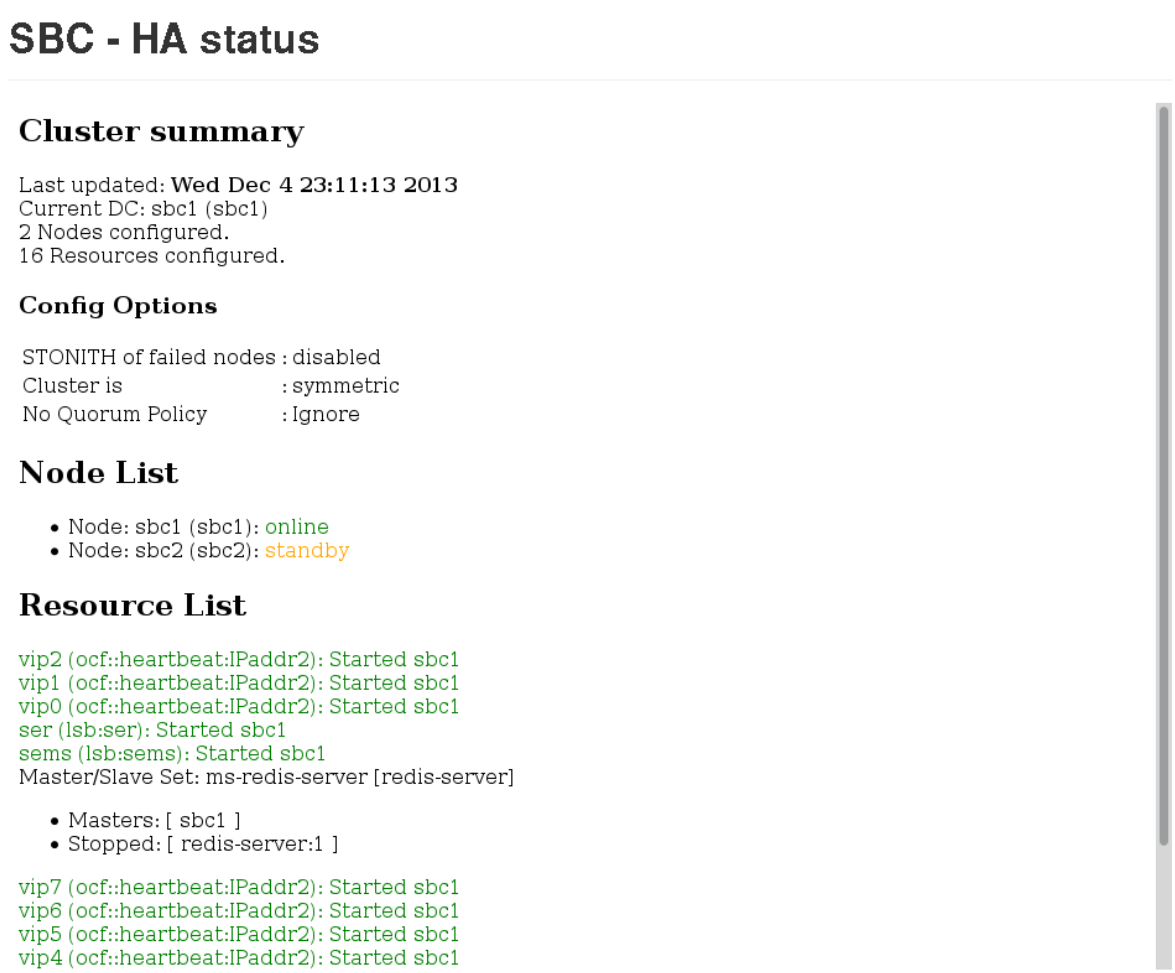
To restore a saved configuration snapshot click on the link *Load* in appropriate row in the list of saved configurations and the configuration will be restored.

Chapter 12

HA

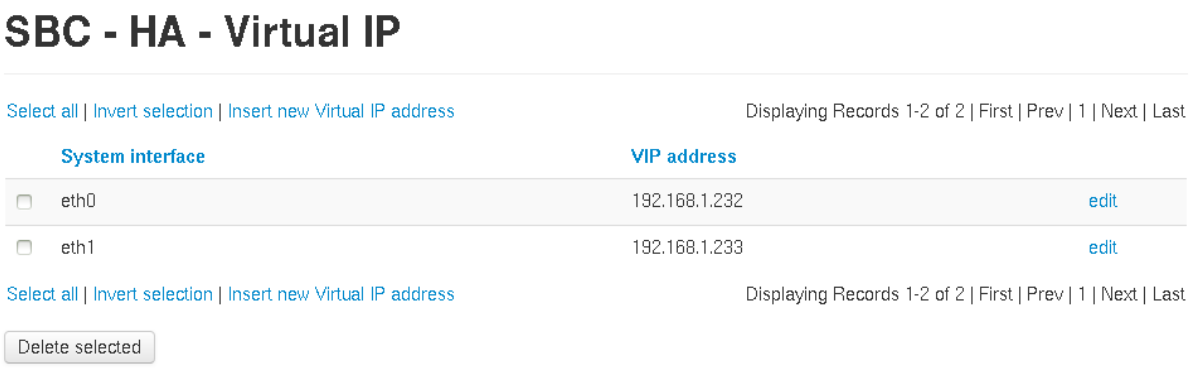
12.1 Status

This screen displays HA status information of a configured FRAFOS ABC SBC or cluster of them including list of configured nodes, resources and activity (active/standby) of the nodes.



12.2 Virtual IP

This screen is used for VIP management. VIPs can be added to system interfaces or removed from them.


SBC - HA - Virtual IP
Select all | Invert selection | Insert new Virtual IP address
Displaying Records 1-2 of 2 | First | Prev | 1 | Next | Last
System interface
VIP address
eth0
192.168.1.232
edit
eth1
192.168.1.233
edit
Select all | Invert selection | Insert new Virtual IP address
Displaying Records 1-2 of 2 | First | Prev | 1 | Next | Last
Delete selected