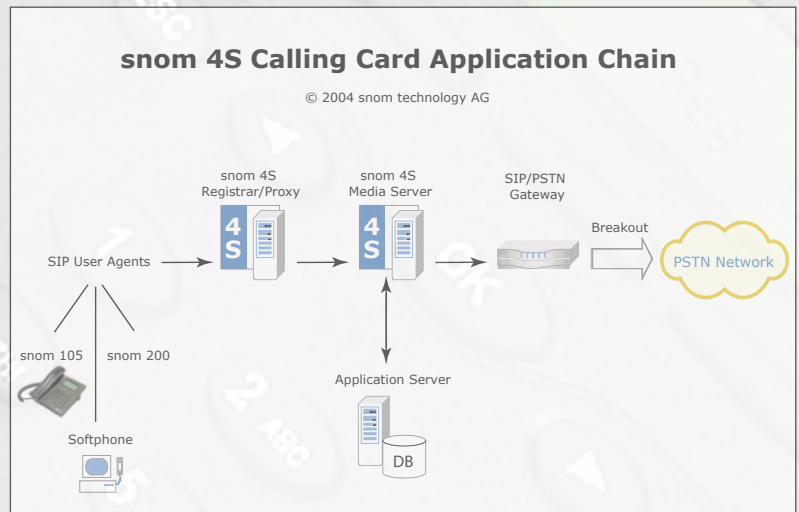


Administrator's Manual



snom 4S

Calling Card System
Version 1.00

snom 4S Calling Card System Version 1.00

Administrator Manual

1. Edition 2004

© 2004 snom technology Aktiengesellschaft. All Rights Reserved.

This document is supplied by snom technology AG for information purposes only to licensed users of the snom 4S Calling Card´s Module and is supplied on an "AS IS" basis, that is, without any warranties whatsoever, express or implied.

Information in this document is subject to change without notice and does not represent any commitment on the part of snom technology AG. The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of that license agreement. It is against the law to copy or use this software except as specifically allowed in the license. No part of this document may be reproduced, republished or retransmitted in any form or by any means whatsoever, whether electronically or mechanically, including, but not limited to, by way of photocopying, recording, information recording or through retrieval systems, without the express written permission of snom technology AG.

Table of Contents

1. Overview	5
2. Usage Scenario.....	7
3. Components	9
3.1. SNOM 4S Proxy/Registrar	10
3.2. SNOM 4S Mediaserver.....	10
3.3. The Application Server	10
<i>The Web Server</i>	11
<i>The Database Server</i>	11
<i>The Application Logic</i>	12
4. Installation.....	13
4.1. Installing the SNOM 4SProxyRegistrar	13
<i>The Proxy Script</i>	13
4.2. Installing the SNOM 4S Mediaserver	18
<i>Creating the Calling Card Account</i>	18
4.3. Installing the Application Server	19
<i>Installing the Web server</i>	20
<i>Installing the Database Server and the Calling Card database</i>	21
<i>Installing the Scripts</i>	25
5. Configuration	27
5.1. Configuring the interface between Proxy/Registrar and Mediaserver	27
<i>Configuring the Dial Plan</i>	27
5.2. Configuring the interface between SNOM 4S Mediaserver and Application Server.....	29
<i>The HTTP GET interface</i>	29
<i>The XML state script</i>	30
6. Administering the Calling Card System.....	33
6.1. Generating new Calling Cards	33

6.2. Setting rates.....	34
6.3. Viewing card or rate information.....	34
Footnotes	35
Index	37

1. Overview

The SNOM 4S Calling Card (S4SCC) Module integrates the SNOM 4S Registrar/Proxy, the SNOM 4S Mediaserver and a back-end application server to provide a fully-fledged calling card system for pre-paid VoIP-to-PSTN telephony.

It provides:

- PIN-based authentication and authorization
- Real-time accounting of calls.
- Customizable Interactive Voice Response (IVR) with full multi-language support¹
- Automatic generation of calling cards
- Web-based administration GUI

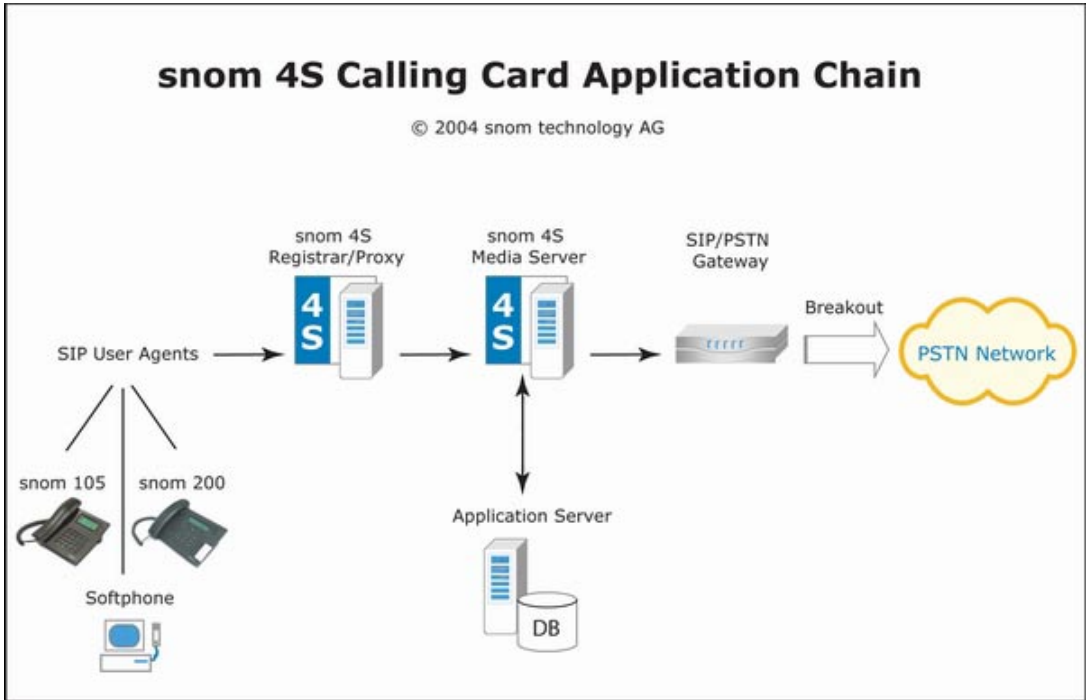
The S4SCC is accessed by registering with the SNOM 4S Proxy/Registrar (S4SPR). The S4SPR handles primary authentication and keeps track of all active users.

At the core of the calling card system lies the SNOM 4S Mediaserver (S4SMS). The S4SMS is a versatile media server, which provides media encoding/decoding as well as sophisticated back-to-back user agent (B2BUA) functionality.

The S4SMS exposes a scriptable state engine, through which the CC service can be adapted to individual needs.

At the end of the application chain lies an SIP/PSTN gateway, which is external to the system. Any RFC3261-compliant SIP/PSTN gateway is suitable.

1



2. Usage Scenario

End-Users of the S4SCC system buy a telephone card with a 12-digit card ID and a secret 4-digit PIN. They access the system with an S4SCC-compliant SIP user agent (SUA).

The client needs to register with the calling card operator's SIP domain, using the card ID as the SIP user name and the PIN as the SIP password.

Example:

Operator's SIP domain: cccall.com
IP address of SIP registrar in charge of cccall.com: 123.123.123.123
Card number: 123456789012
PIN: 1234

Register Command
REGISTER sip:123.123.123.123 SIP/2.0
From: <sip:123456789012@cccall.com>;tag=XXXX
To: <sip:123456789012@cccall.com>

....

Upon receiving this REGISTER request, the registrar sends back an

SIP/2.0 407 Proxy Authentication Required

The user enters his password (alternatively, he may have preconfigured it with his SUA) and the SUA re-sends the REGISTER request with the required authentication information. If the authentication information matches the database of valid and active calling cards, the registrar sends back a

SIP/2.0 200 OK

Via:....

From: <sip:123456789012@cccalls.com>;tag=XXXX

To: <sip:123456789012@cccalls.com>

X-SNOM-CCTOKEN: 98765432

This response contains a proprietary “token”², which provides the SUA with a secure way of accessing the S4SCC without needing to re-authenticate at each request. The token is valid for the duration of the registration. Upon re-registering, a new token is issued. The client needs to be compliant with this protocol extension.

Once the registration sequence described above has been completed, the user can place calls, which will be charged to his calling card.

The S4SCC application server backend contains a rate table that determines which country and city codes can be accessed and how much they cost. Each calling card is assigned to one rate table in order to allow the operator to offer different cards with different pricing models.

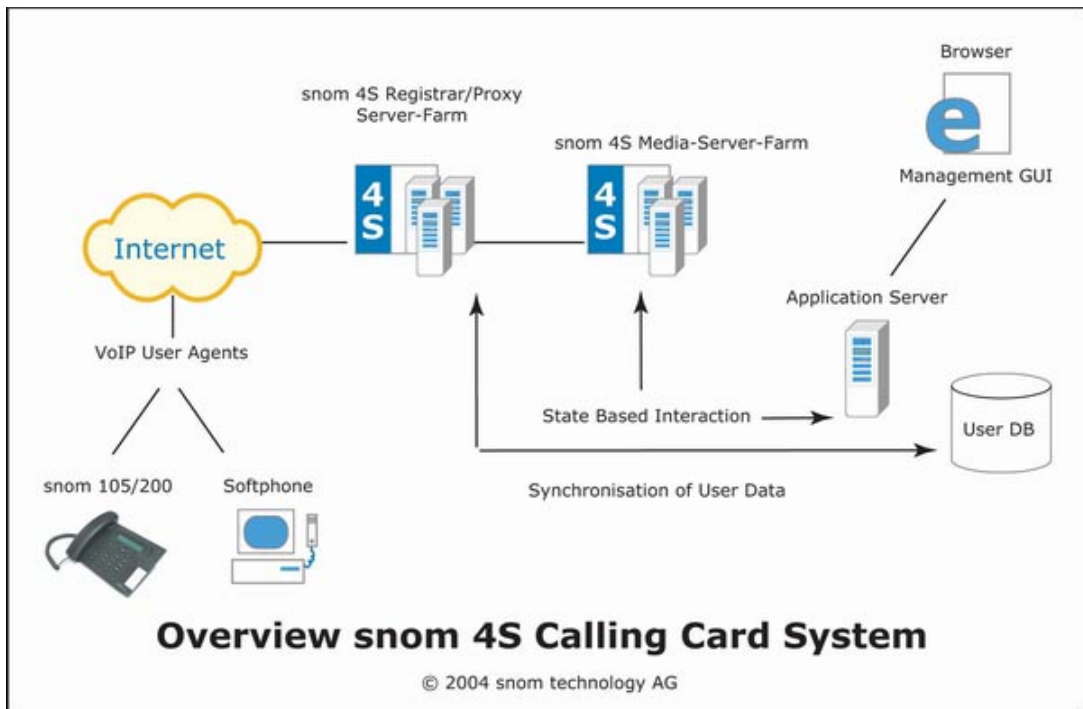
The user places a call by dialing the telephone number of his (PSTN) counterpart. The S4SCC will verify that this is a reachable destination and establish the connection via the SIP/PSTN breakout gateway.

Accounting starts after the media connection between SUA and PSTN counterpart has been established.

The SNOM 4S Mediaserver at the core of the system goes through a state transition after each billing period. Based on this transition, the accounting database is updated and real-time balance information is sent to the SUA³. All accounting and billing is based on this real-time accounting information.

3. Components

The main components have already been described. The following image shows their interaction.



3.1. SNOM 4S Proxy/Registrar

The S4SPR acts as the authentication and authorization gateway to the system. Its user information needs to be synchronized with the user database in the application server.

It forwards all authenticated calls to the media server, which is configured through the dial plan. The token-based authentication/authorization is configured using a proxy script.

The actual settings and scripts will be described in the “Configuration” section.

3.2. SNOM 4S Mediaserver

The calling card system core component is a special B2BUA within the media server⁴. It is based on the model of a state machine, which can be scripted through a simple XML-based configuration language.

Although it is based on a state machine model, it is stateless in that it keeps no call state in its memory space⁵.

All call state is kept in the application server (AS; see below). The Mediaserver interacts with the AS through simple HTTP GET requests. It initializes state for each call through a request to

`http://<application_server>/cc/cc_init.php.`

This request generates a session ID and passes it back to the Calling Card B2BUA. This session ID is used in all interactions with the application server.

3.3. The Application Server

The application server consists of:

1. A web server (Apache)
2. A database server (MySQL)

3. The application logic (written in PHP and Perl)

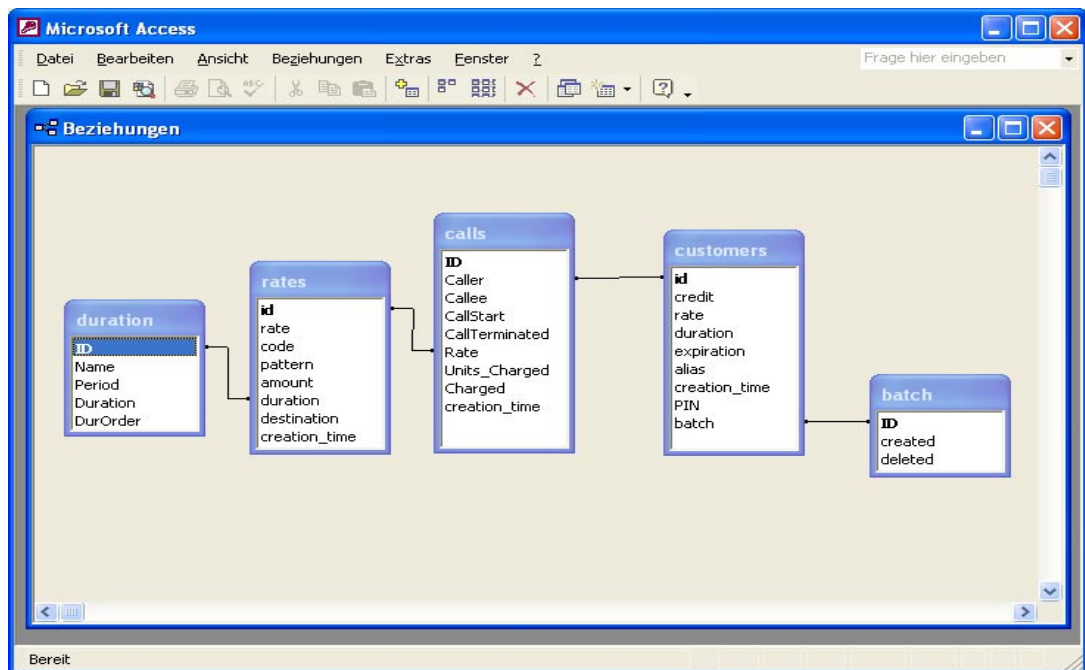
The Web Server

We are currently using Apache 2.0.48 with SSL support. No other special features are needed.

The Database Server

We are currently using MySQL 4.0.16. No special features are needed.

The database itself is quite simple, with an ERD given below:



The Application Logic

Most of the application logic is written PHP 4. For some batch utilities we use Perl 5. The main scripts are:

1	cc_init.php	Initializes state for one call. Validates card number and destination number. Returns: session ID
2	cc_rate.php	Sets the most important parameters for the calling card module: balance, credit, rate, period As a side effect, instantiates the main server side objects (cc_call and cc_customer)
3	cc_charge.php	Charges the card one unit at the set rate. Accounting is done in real time by writing to the DB Returns current balance
4	cc_call_end.php	Writes CDR to database. Ends call.
A	cc_generate_cards.php	Generates a new batch of calling cards and sends result as ("Card ID";"PIN") to Email specified
B	cc_import_cards.php	Imports cards from a file
C	cc_show_cardInfo.php	Displays Calling Card information
D	cc_import_rate.php	Imports rate information from a file
E	cc_show_rate.php	Displays rate information

Pages 1-4 are used to interact with the Mediaserver.

Pages A-E are used to administer the system.

4. Installation

Installation consists of the following main steps:

- Install the SNOM 4S Proxy/Registrar
- Install the SNOM 4S Mediaserver
- Install the Application Server
 - Install the Web server
 - Install the Database Server
 - Import the Callingcard database
 - Include the PHP scripts to a directory served by the Web server

4.1. Installing the SNOM 4S Proxy Registrar

There are no special installation procedures beyond the standard SNOM 4S Proxy/Registrar installation.

Please refer to the SNOM 4S Proxy/Registrar documentation.

Once installed, you will need to set up the SIP domain for the CC system and configure your DNS accordingly. We will assume the domain name cccall.com.

Please also verify that the Proxy script needed for the "X-SNOM-CCTOKEN" extension is included.

The Proxy Script

The following proxy script needs to be set:

```

user_directory($user)
{
    return rightstr($user,3);
}

on_register($user, $nat, $t) {
    if ( $nat == "" ) {
        $cctoken = random(100000,400000);
        register($user,$t, $nat,$cctoken);
        if ( registered($user) )
        {
            set_parameter($user, "cctoken", $cctoken);
        }
    } # of nat
    else {
        if ( !$reject_nat_register ) {
            if ( $ka_regtime == "" ) { $t = 20; }
            else {
                $t = $ka_regtime;
            }
            $cctoken = random(100000,400000);
            set_parameter($user, "cctoken", $cctoken);

            register($user,$t, $nat,$cctoken);
            if ( registered($user) )
            {
                set_parameter($user, "cctoken", $cctoken);
            }
        } # if !reject_nat
        else {
            reject_request("406 Bad Contact (NAT)");
        }
    }
}

on_request() {
    $user = parse_user($request_uri);

    if ($account != "" && get_parameter($account, "disabled")) {
        # someone disabled this account
        reject_request("402 Account Disabled"); # default
    }
    else if (parse_scheme($request_uri) == "tel") {
        # convert the telephone number into an enum suffix

```



```

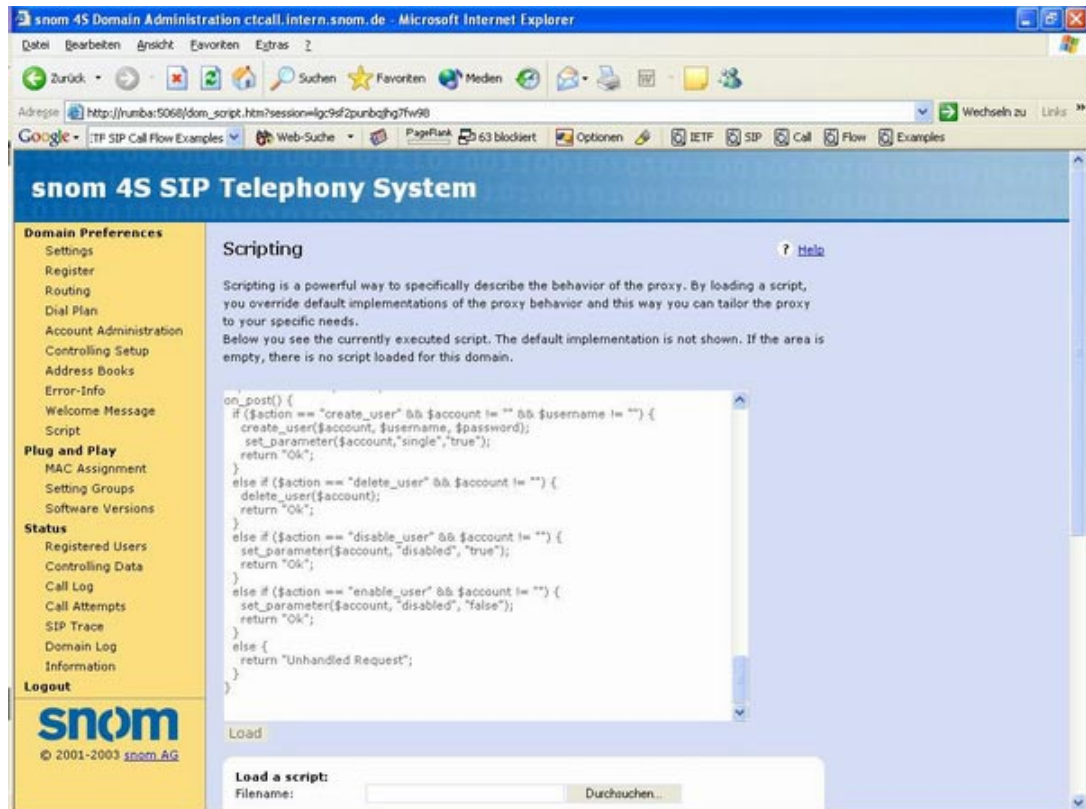
    else {
        reject_request("404 Not Registered");
    }
} # other methods
} # known user
else {
    # no for a known user:
    if (!exec_dialplan(parse_url($from), $request_uri, $request_uri, $allow_
pattern)) {
        reject_request("404 Unknown User"); # default
    }
}
} # if METHOD == INVITE
} # if tokens matched
else
{
    reject_request("403 Forbidden invalid OSP Token");
}
} # else speed dial
} # on_request

# handle posts to the proxy domain, set the variables to the values
# provided in the post request.
on_post() {
    if ($action == "create_user" && $account != "" && $username != "") {
        create_user($account, $username, $password);
        set_parameter($account,"single","true");
        return "Ok";
    }
    else if ($action == "delete_user" && $account != "") {
        delete_user($account);
        return "Ok";
    }
    else if ($action == "disable_user" && $account != "") {
        set_parameter($account, "disabled", "true");
        return "Ok";
    }
    else if ($action == "enable_user" && $account != "") {
        set_parameter($account, "disabled", "false");
        return "Ok";
    }
    else {
        return "Unhandled Request";
    }
}
}

```

You can verify that the script is included by going to the “Domain Preferences”->“Script” page of your SIP domains administration page on the S4SPR.

If the script is not set, you can copy the script included above and paste it into the text window. Then press “Save” and the script should be loaded.



4.2. Installing the SNOM 4S Mediaserver

Creating the Calling Card Account

Make sure you have a Callingcard license. You can verify this by looking at your license string, which should start with the string "snom-media-cc1".

Install the SNOM 4S Mediaserver as per its documentation.

Verify that the Calling Card state script is in the correct location. It should be in

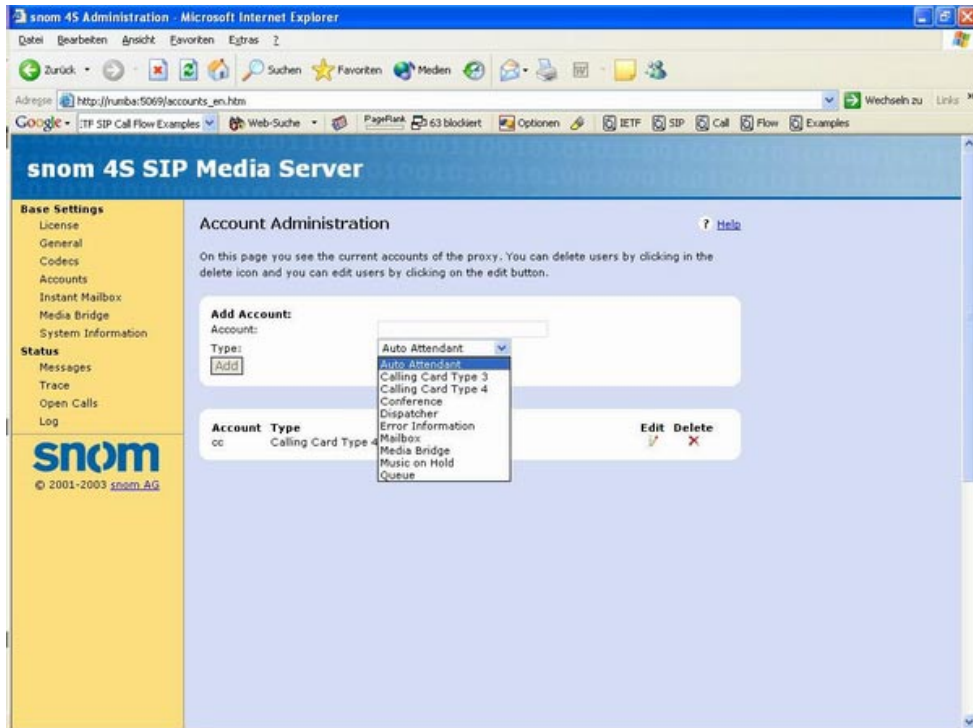
<Your Mediaserver's runtime directory>/xml/callcard.xml

You can verify your Mediaserver's runtime directory by checking its start-up configuration file (/etc/rc.config on SuSE Linux, /etc/mediaserver.conf on RedHat).

Once you have verified this, go to the main administration page of your Mediaserver.

- Click on Accounts.
- Type in the name of the account. We will assume "cc".
- Select "Calling Card Type 4" from the "Type" pulldown.
- Click on "Add"

This creates the CC B2BUA on your SNOM 4S Mediaserver. This account is the account addressed by the S4SPR's dial plan (described below).



You will also need to provide an SIP subdomain for the Mediaservers and configure your DNS accordingly. We assume a domain name of ms.cccall.com.

4.3. Installing the Application Server

The Application Server consists of

- A Web server:
- A Database server: MySQL 4.0.16
- The Calling Card database
- Application logic in the form of PHP4 and Perl5 scripts

You will also need cURL, a command-line handling URL utility. This can be downloaded from <http://curl.haxx.se/> .

In this section we will assume both the web server and the database server to have been installed and running.

Here, we will only describe the installation steps that are specific to the S4SCC system.

Installing the Web server

We recommend using Apache (version 2.0.x). Please refer to www.apache.org for installation instructions of the Apache web server.

However, any Web server capable of serving PHP4 scripts will be suitable.

Apache is configured through a configuration file called `httpd.conf`.

The following excerpt shows the only settings specific to the S4SCC:

```
LoadModule php4_module modules/libphp4.so
:
:
Alias /cc/ "/ourPHPDirectory/"
:
:
User snom
```

These three settings specify:

1. That Apache loads the PHP4 module to interpret PHP4 scripts.
2. The directory, which contains the PHP scripts (cc_init.php etc.).
3. That Apache should run under the User ID snom. Any user other than Nobody will do.

The reason for the second setting is that some scripts need to send out Emails and on many UNIX systems, user Nobody is barred from doing so.

Installing the Database Server and the Calling Card database

We recommend using MySQL version 4.0.x. However, any SQL-compliant database for which a PHP4 driver is available will be suitable.

The only steps specific to S4SCC are:

- Create the Calling Card database: it should be called "Callingcard"
- Add one user who has full table access rights. In our example we will call the user "cccall" with a password of "cccall". This will be the user that the PHP4 scripts will use to access the database
- Create the database according to the ERD given above.

The SQL code to create the database tables is:

```
#
# Host: localhost
# Time: 20. Januar 2004 um 13:42
# Server Version: 4.0.16
# PHP-Version: 4.3.3
#
# Database: `callingcard`
#

# -----

#
# table structure for table `batch`
#

DROP TABLE IF EXISTS `batch`;
CREATE TABLE `batch` (
  `ID` int(11) NOT NULL auto_increment,
  `created` timestamp(14) NOT NULL,
  `deleted` timestamp(14) NOT NULL,
  PRIMARY KEY (`ID`)
) TYPE=MyISAM COMMENT='Batch table to separate
statistically generated card numbers' AUTO_
INCREMENT=84 ;

# -----

#
# table structure for table `calls`
#

DROP TABLE IF EXISTS `calls`;
```

```

CREATE TABLE `calls` (
  `ID` double unsigned NOT NULL default '0',
  `Caller` varchar(12) NOT NULL default '',
  `Callee` varchar(20) NOT NULL default '',
  `CallStart` datetime NOT NULL default '0000-00-00 00:
00:00',
  `CallTerminated` datetime NOT NULL default '0000-00-
00 00:00:00',
  `Rate` varchar(32) NOT NULL default '',
  `Units_Charged` int(11) NOT NULL default '0',
  `Charged` float NOT NULL default '0',
  `creation_time` timestamp(14) NOT NULL,
  UNIQUE KEY `ID_3` (`ID`),
  KEY `Rate` (`Rate`)
) TYPE=MyISAM COMMENT='CDR Table';

```

```
# -----
```

```
#
# table structure for table `customers`
#
```

```

DROP TABLE IF EXISTS `customers`;
CREATE TABLE `customers` (
  `id` decimal(12,0) NOT NULL default '0',
  `credit` double default NULL,
  `rate` varchar(32) default NULL,
  `duration` varchar(10) default NULL,
  `expiration` varchar(14) default NULL,
  `alias` varchar(255) default NULL,
  `creation_time` varchar(14) default NULL,
  `PIN` varchar(4) NOT NULL default '',
  `batch` int(11) NOT NULL default '0',
  PRIMARY KEY (`id`),
  KEY `rate` (`rate`)
) TYPE=MyISAM;

```

```
# -----
```

```
#
# table structure for table `duration`
```

```

#
DROP TABLE IF EXISTS `duration` ;
CREATE TABLE `duration` (
  `ID` int(10) unsigned NOT NULL auto_increment,
  `Name` varchar(12) NOT NULL default '',
  `Period` int(10) unsigned NOT NULL default '60',
  `Duration` int(11) NOT NULL default '-1',
  `DurOrder` int(10) unsigned NOT NULL default '0',
  PRIMARY KEY (`ID`),
  UNIQUE KEY `Name` (`Name`)
) TYPE=MyISAM COMMENT='This table is used for variable
billing purposes' AUTO_INCREMENT=2 ;

# -----

#
# table structure for table `rates`
#

DROP TABLE IF EXISTS `rates` ;
CREATE TABLE `rates` (
  `id` int(11) NOT NULL auto_increment,
  `rate` varchar(32) default NULL,
  `code` varchar(11) default NULL,
  `pattern` varchar(255) default NULL,
  `amount` double default NULL,
  `duration` varchar(12) NOT NULL default 'default',
  `destination` varchar(40) default NULL,
  `creation_time` timestamp(14) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `rate` (`rate`)
) TYPE=MyISAM AUTO_INCREMENT=1861 ;

```

Installing the Scripts

Installing the PHP4 scripts and the web GUI scripts is easy: just move the contents of the cc directory to the directory configured as the web server's alias for "/cc/" (see section 4.3)

Move the contents of the Perl5 scripts to a directory of your choice. Let us assume /home/cccall/ccperl.

In the PHP script "cc_settings.inc" you will now need to set a few variables to configure your environment.

Example settings are:

```
function cc_settings()
{
    $this->db_path = "localhost";
    $this->db_user = "cccall";
    $this->db_pass = "cccall";
    $this->db_name = "callingcard";
    $this->gateway["natl"] = "192.168.0.1";
    $this->gateway["intl"] = "192.168.0.1";
    $this->perl = "/usr/bin/perl";
    $this->perldir = "/home/cccall/ccperl";
}
```

The two entries for `$this->gateway` point to the:

- SIP/PSTN gateway for international calls (numbers beginning with "00") at `$this->gateway["intl"]`

- Default SIP/PSTN gateway at `$this->gateway["natl"]`

The PERL scripts also need to be configured. In the `ccperl` directory in which the scripts are located, you will have to edit the script `settings.pl`. An example script is shown below:

```
$applicationServer = "demawend";
$MAIL_CMD = «/usr/bin/mail»;
$domain=»cccall.com";
$domainPass="cccall";
$CURL = "/usr/local/bin/curl --stderr /dev/null";
$mysql_CMD = "/usr/local/bin/mysql -N -n ";

$proxyhosts[0] = "http://rumba:5068/post.htm";
```

The first line specifies the hostname or IP address of the application server.

`$domain` is the name of your SIP domain.

`$domainPass` is the login password for the Web administration interface of that domain.

`$proxyhosts` is an array specifying the IP address(es) of your SNOM 4S proxy/registrar(s). As Redundancy is not covered in this manual, we will limit ourselves to the discussion of a single proxy installation.

5. Configuration

Configuring the S4SCC essentially involves setting up the individual components to interact with each other.

This means configuring the following interfaces:

- S4SPR <-> S4SMS
- S4SMS <-> Application Server

5.1. Configuring the interface between Proxy/Registrar and Mediaserver

The only thing you need to configure on the proxy is the Dial Plan.

Configuring the Dial Plan

This is relatively straightforward. You need to configure this in order to forward all calls from calling card users, which go to a (PSTN) telephone number to the Calling Card account on the Mediaserver.⁶

The Dial Plan consists of the following fields:

Mode, From, To, Argument

Mode in our case is always Forward.

From is the originating SIP URI, in our case any SIP URI with a 12-digit user ID and our domain.

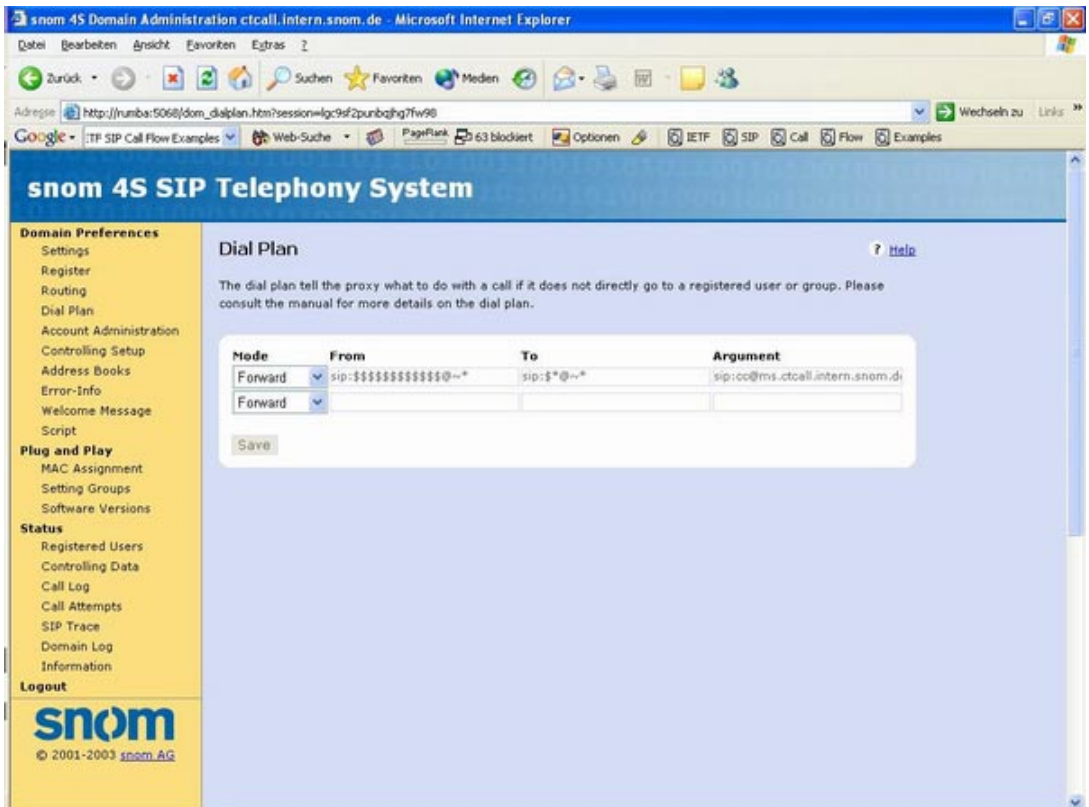
To is the SIP URI being called, in our case any user ID consisting only of digits

Argument: in the case of Forwarding, this is the SIP URI to which the call is forwarded. In our case this corresponds to the SIP URI of the Calling Card B2BUA on the Mediaserver.

Example (assuming your SIP domain is cccall.com and the Mediaserver's domain is ms.cccall.com):

Mode: Forward
 From: sip:\$\$\$\$\$\$\$\$\$\$\$\$@~*
 To: sip:\$*@~*
 Argument: sip:cc@ms.cccall.com

Please also refer to the following screen shot:



5.2. Configuring the interface between SNOM 4S Mediaserver and Application Server

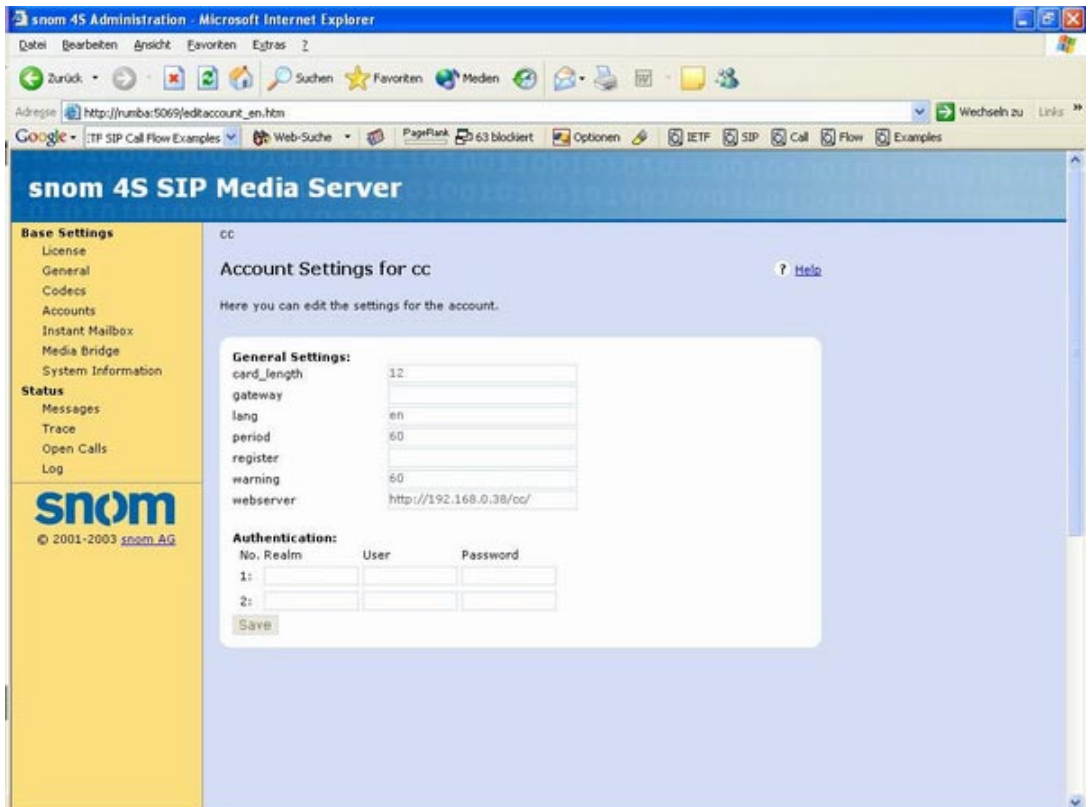
Two components define the interaction of the Mediaserver and the Application Server:

1. The HTTP GET interface
2. The XML script for the B2BUAs state machine

The HTTP GET interface

The S4SMS communicates with the application server via HTTP GET requests. You will therefore need to point the Mediaserver's CC B2BUA to the URL of your application server. This is done by editing the Mediaserver's CC account.

- Go to the main administration page of your Mediaserver.
- Click on "Accounts"
- Select the account you created as per section 4.2
- Click on "Edit".
- Enter the URL of your application server in the field "webserver".
- Click on "Save".



5

The XML state script

For a general description of the XML-based scripting language, please refer to the SNOM 4S Media Server XML Scripting guide.

We include an excerpt of a reference script to show how the XML script is used to execute state-dependent HTTP GET requests.

For instance, refer to the state with the name "direct_call". This is the initial state of a reference installation.

You will see that it sets a variable called `account` to the value `$f_un`.

This means that the internal variable `account` is set to the SIP user ID from the `From:` header.

The second important thing to note is the line that reads:

```
web_get webserver/cc_init.php?account=$account&destination
=$collect
```

This calls the script `cc_init.php` with the HTTP parameter `account` set to the value of the internal variable `account`, and the HTTP variable `destination` set to the value of the internal variable `collect`. This variable contains the telephone number dialed by the user.

```
<state name="direct_call">
<event name="enter">
<cmd>set account $f_un</cmd>
<cmd>set collect ${t_un:get_field:0:}</cmd>
<cmd>
web_get webserver/cc_init.php?account=$account&de
stination=$collect</cmd>
<cmd>goto wait_for_session</cmd>
</event>
</state>
<state name="wait_for_session">
<event name="web">
<cmd>set session $session</cmd>
<cmd>web_get $webserver/cc_rate.php?PHPSESSID=$session</
cmd>
<cmd>message "snom Calling Card System"</cmd>
<cmd>goto wait_for_rate</cmd>
</event>
</state>
<state name="wait_for_session">
<event name="web">
<cmd>set session $session</cmd>
<cmd>web_get $webserver/cc_rate.php?PHPSESSID=$session</
cmd>
<cmd>message "snom Calling Card System"</cmd>
<cmd>goto wait_for_rate</cmd>
</event>
</state>
```

5

6. Administering the Calling Card System

The main administration tasks for the CC system are:

- Generating new Calling Cards
- Setting rates
- Viewing card or rate information

6.1. Generating new Calling Cards

New calling cards can be generated by going to

http://<your_application_server>/cc/cc_generate_cards.php

You will need to enter an Email address to which the resulting calling card IDs and their PINs are sent. Please also note that you shouldn't generate more than 10,000 cards in one batch.

You also have to select which rate you want to use for that batch of calling cards (see below for generating rates).

The generated Calling Card IDs are of the format "1" + "limit Type" + 10-digit random component.

Limit Type is either "0" or "1". Currently this means that a card of limit type "0" will have a credit of "10" and a card of limit type "1" will have a credit of "20".

In future versions of the S4SCC this will be freely configurable through a web interface.

6.2. Setting rates

You need to provide rate information in CSV-Format.

The file contains the international prefix, the national prefix, the name and the price each separated by a semicolon. If a number cannot be reached via a rate table entry, the calling card will not be able to place a call to that number. All rates are measured at 60-second intervals.

The file has the following columns, separated by a semicolon (";"):

International Prefix;National Prefix;Name of Rate;Cost per minute

Example:

```
0049;17;GERMANY;0.5
```

```
0049;30;GERMANY;0.2
```

This example creates a rate called "GERMANY" with two entries, one for all numbers starting with "004917", charged at 0.5 per minute, and one for "004930" charged at 0.2 per minute.

6.3. Viewing card or rate information

Rates can be viewed at

```
http://<your_application server>/cc/cc_show_rates.php
```

Calling Card Information can be viewed at

```
http://<your_application server>/cc/cc_show_cardInfo.php
```

Footnotes

¹ Currently German, English, Japanese and Chinese (Mandarin)

² The token is passed in a special header („X-SNOM-CCTOKEN“) in the „200 OK“ response to the REGISTRATION request.

³ As an SIP MESSAGE

⁴ A special license is needed to enable this feature.

⁵ The fact that it is stateless is relevant for load-balancing purposes.

⁶ For detailed instructions about the dial plan, please refer to the SNOM 4S Proxy/Registrar Administrator's Guide

⁷ This is one of the prefixes used for Germany's mobile networks

⁸ This is the prefix for Berlin.



Index

A

Administering the Calling Card System 33
Apache 10
Application Logic 12

B

B2BUA 10
billing 8

C

Calling Card Account 18
Components 9
 4S Proxy/Registrar 10
 SNOM 4S Mediaserver 10
 The Application Server 10
Configuration 27
 interface between Proxy/Registrar and Mediaserver 27
 interface between SNOM 4S Mediaserver and Application Server 29
Configuring the Dial Plan 27

D

Database Server 11

E

End-Users 7

G

Generating new Calling Card 33

H

HTTP GET interface 29

I

Installation 13
Installing the Application Server 19
Installing the Calling Card database 21
Installing the Database Server 21
Installing the Scripts 25
Installing the SNOM 4S Mediaserver 18

Installing the SNOM 4S **V**
ProxyRegistrar 13

Installing the Web server 20

34

Viewing card or rate information

M

MySQL 10

W

Web Server 11

O

Overview 5

X

XML script 29

XML state script 30

P

Proxy Script 13

PSTN counterpart 8

S

Setting rates 34

SIP password. 7

SUA 7

T

token 8

U

Usage Scenario 7

Reader's Feedback

snom technology AG welcomes your evaluation of this manual and any suggestions you may have. These help us to improve the quality and usefulness of our documentation.

Please send your comments and suggestions to:

snom technology AG

Attention: Marketing Department

Pascalstr. 10B, 10587 Berlin, Germany

FAX: +49 (30) 3990 7839

Manual Name: Administrator's Manual, snom 4S Calling Card System

Software Version: 1.00 Date: February 2, 2004

	Excellent	Good	Fair	Poor
How would you rate the manual overall?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the installation instructions effective?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the Configuration Instructions effective?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Is the manual properly organized?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the illustrations usefull and easy to understand?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Are the suggested and default values useful?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Is the index useful?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Did you find any errors in the manual? (Please reference page and paragraph)_____

How might we improve this manual?_____

Name_____ Title_____

Company Name_____ Telephone ()_____

Thank you for taking time to fill out this form.



snom 4S Calling Card System
Administrator Manual
2 Feb, 2004, Version 1.00
© 2004 snom technology AG
All rights reserved.
Subject to change without notice.

Europe & ROW:

snom technology AG
Pascalstr. 10B
10587 Berlin, Germany
Phone: +49 (30) 39833-0
mailto:info@snom.com
http://www.snom.com
sip:info@snom.com

USA and Americas:

snom USA Representation
ABP International, Inc.
1203 Crestside Dr.
Coppell, Texas 75019, USA
Phone: +1-972-831-0280
sip:usa@snom.com
mailto:usa@snom.com

India and SAARC:

snom technology (India) Pvt Ltd.
No. 417, International Trade Tower
Nehru Place, New Delhi-110019
Phone: +91 11 26234097
Fax: +91 11 26234079
http://www.snomindia.com
mailto:info@snomindia.com
sip:india@snom.com