« *networking the networkers* »

TERENA SIP Handbook

A practical guide to setting up a safe VoIP and videoconferencing server: the NREN-Enhanced Communications Server or N-ECS

Written by the TERENA Task Force on Enhanced Communication Services – TF-ECS

September 2008

www.terena.org/tf-ecs

Trans-European Research and Education Networking Association

Acknowledgements

Contributors to the NREN-Enhanced Communications Server:

George Fergadis (AUTH)
Faidon Liambotis (GRNET)
Jan Růžička (CESNET)
Lukáš Macura (CESNET)
Mihály Mészáros (NIIF)
Mina Angelopoulou (UoA)
Miroslav Dvořák (CESNET)
Niels van Dijk (SURFnet)

Other TF-ECS members:

Dimitris Daskopoulos (GRNET)
Erik Dobbelsteijn (SURFnet/TWIYO) – Co-Chair
Licia Florio (TERENA)
Bernie Hoeneisen (SWITCH)
András Kovács (NIIF)
Olav Kvittem (UNINETT)
Cătălin Meiroşu (TERENA)
Harri Salminen (CSC)
Bo Ståhle (UNI-C)
Kewin Stoeckigt (AARnet)
Marco Somani (GARR)
Péter Szegedi (TERENA)
Fabio Vena (SWITCH) – Co-Chair

Many thanks to the following contributors to TF-ECS:

João Pereira (FCCN)
John Martin
José M. Fontanillo (RedIRIS)
Luís Guido (FCCN)
Michiel Schok (SURFnet)
Milivoie Mirovic (AMRES)
Rui Ribeiro (FCCN)
Slayko Gaiin
Simon Horne (Packetizer)
Valentino Cavalli (TERENA)
David Vrtin (ARNES)
Yannis Mitsos (GRNET)

Edited by Erik Dobbelsteijn

Foreword

Goal

This *TERENA SIP Handbook* describes Voice over Internet Protocol (VoIP) and videoconferencing technologies, and how to set them up and test them using the NREN-Enhanced Communications Server (N-ECS). The N-ECS is, in fact, a virtual machine image that contains both basic as well as advanced (IP) telephony and videoconferencing features on one machine. It provides guidelines and information about the IP telephony world and recipes to build up an experimentation server and its goal is to provide the user community, mainly in National Research and Education Networks (NRENs) and connected institutes, with an easy way to get started quickly.

Since this handbook is intended to be a technical document, the main target audience consists of network engineers and system administrators. However, university students and researchers may find it useful, both for enriching their technological background and for finding information about advanced research topics and projects in the European community.

A first edition of the *IP Telephony Cookbook* was published by TERENA in 2004. The content is available online at http://www.terena.org/activities/iptel/contents1.html. It had a slightly different starting point, namely to cover as much available open source software as possible, therefore resulting in many recipes. The evolution of technology and the continuous interest of the NREN community in this topic called for an update to the technology descriptions that were in the old cookbook. But more importantly, it was necessary to get interested people started more easily and to provide them with the means to progress to more advanced configurations more quickly. So, being more than an update of the old cookbook, this new handbook is dedicated to the NREN-Enhanced Communications Server, which is mainly based on Session Initiation Protocol (SIP).

Reasons for writing this document

Recommendations for setting up IP telephony solutions at university- and national-level, with information about protocols and the interoperability of equipment, are widely available on the Internet. However, the viewpoints are fragmented and usually focus on particular scenarios. This may be a result of commercial interest, or because they are only used within a private IP cloud, or simply because they describe a particular solution. A good starting point is offered by the Internet2 SIP.edu project (http://www.internet2.edu/sip.edu/). Still, it is very difficult to set up a feature-complete system that combines the most powerful roles of various open-source systems. More importantly, little has been done to put such scenarios in a wider context, taking into account multi-domain aspects. Mainly for these reasons, a number of people in the TERENA community with significant expertise in the area of IP telephony decided to undertake this task in 2007 and to compose the virtual machine image and this corresponding document.

Contents

The *TERENA SIP Handbook* is divided into chapters that guide the reader through increasing levels of knowledge of the IP telephony world in general and the set-up of the NREN-Enhanced Communication Server in particular. This first chapter contains introductory information and gives details of the contents and useful tips on how to read this document. Some techno-economic considerations may be found in the introductory section of the preceding edition of the *IP Telephony Cookbook*. It is perhaps not surprising that most of these reflections were still valid in 2008, when this handbook was produced.

Chapter 1 explains the technological background needed in order to understand the topics addressed in the rest of the handbook. As this type of basic information is widely available online, the authors decided to mainly include references for further reading. The articles and documents referenced were chosen for their technical accuracy, with a preference for hosting sites that were expected to be long lasting. After the basics of IP telephony are described, Chapter 2 sketches the main scenarios in which IP telephony can be deployed. It is important to realise in what contexts a telephony server will be deployed and what services it will provide, in order to follow the right deployment steps.

Chapter 3 explains how to prepare to set up your own NREN-Enhanced Communications Server. Please read those steps carefully before proceeding. Chapter 4 then details the basic steps involved in setting up IP telephony services inside an administrative domain. Advanced topics, such as peering across domains and value-added services, are introduced later in this chapter. Finally, Chapter 5 presents technical details about the IP telephony services deployed by European NRENs and lists options for their interconnection. The live status can be found on the TERENA website (http://www.terena.org/activities/n-ecs/).

Way of working

The server software and tools are mostly available through open source licences or are free to obtain and known for their compliance to open standards. The decision about which to use is solely based on the experience of the contributors and by no means represents exhaustive investigations into the options available on the market. To cover the main reason for starting this work, namely to safely interconnect IP telephony 'islands' through VoIP (and not Integrated Services Digital Network (ISDN)), the contributors had little choice but to use these pieces of software.

To access the knowledge obtained by and available within the community of contributors in more depth, NRENs or institutes looking into new deployments of a particular tool can contact the contributors.
By maintaining the downloadable NREN-Enhanced Communications Server virtual server image, the experimenter is always provided with up to date technology and the latest features to test.

How to read this document

Since the *TERENA SIP Handbook* is a technical reference document, it includes guidelines for users who do not want to read the entire document so that they can find the information they need. In this section, we give the reader tips on how to read the document in order to retrieve the information needed as quickly as possible; for a detailed overview of the contents of the handbook, please refer to the previous section. To speed up the information retrieval process, each reader should identify himself as belonging to one of the following three groups:

- readers who have no knowledge of IP telephony;
- readers who have a basic knowledge of IP telephony;
- readers who have advanced knowledge of IP telephony.

Readers belonging to the first group should, first of all, consult Chapter 2 of the preceding edition of the *IP Telephony Cookbook* to acquire the necessary background. Readers who are interested in setting up an IP telephony service should read Chapter 2 of this handbook to get a clear picture of the possible scenarios offered by IP telephony and to target the one best suited to their needs. The second and third groups of readers may access the handbook at will, depending on the type of problem for which they are looking for a solution. It should be noted by users in the second group that the inter-domain aspects involved in the last part of Chapter 2 may require a deeper understanding of the technical aspects involved. Those actually performing tests using the N-ECS should definitely read Chapter 4, which contains the information necessary for making preparations.

The following graphical hints are used to point out when to use a Linux command or a configuration file item:

```
Configuration items are written in Courier New
```

`Linux commands are written in Courier New bold`

All three groups of users may find useful information in Chapter 5 regarding other deployments within the academic community.

Licensing

The N-ECS image and its documentation collectively, further referred to as 'the work', were assembled with great care. The intention is to provide the technical community world-wide with in-depth information with working examples, free of obligation. There is no other intention, such as providing paid and / or production-ready products or services. Neither TERENA nor its members or any contributors to this work can be held responsible for any consequences of the use of this work.

The work can be distributed freely when unchanged. Any modifications should be annotated clearly and reported back to TERENA and the original contributors should always be mentioned.

All components used for the N-ECS are subject to their respective licences, please read and follow them carefully.

Contents

Contents (continued)

1 Theoretical background

This chapter illustrates common issues and explains terms and abbreviations, putting them in the context of the NREN-Enhanced Communications Server (N-ECS).

1.1 Creating an island

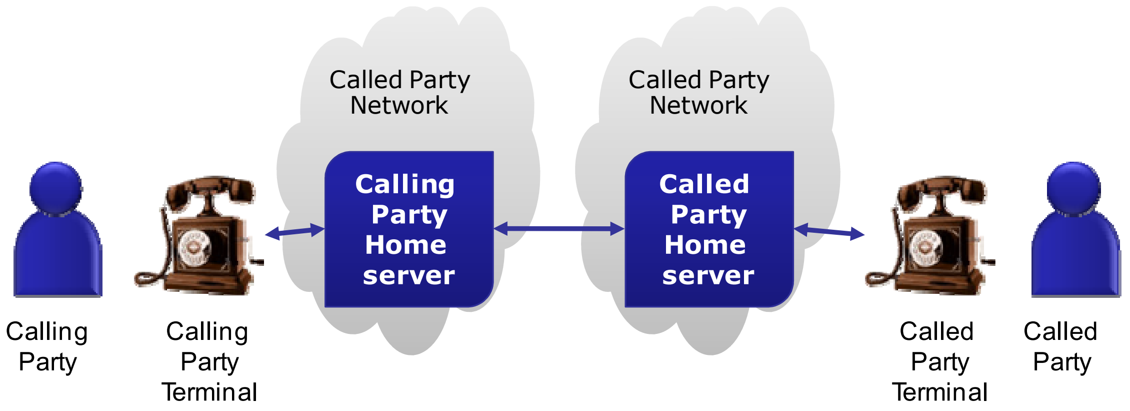
It has become fairly easy to set up a simple VoIP server, connect VoIP phones to it and connect it to an ISDN line. However, such an implementation is an island that can only be reached through the traditional telephony network, using audio only. The advantages of VoIP are only experienced within the organisational context, but cannot be used in connection with the outside world. Slowly, the promise of Internet-based calling is coming nearer, with providers offering voice connections over IP called 'SIP trunking'. Unfortunately, calls are usually still restricted to going through these providers only. This protects their business, but also enables them to control the feature set and quality. Fortunately, the island can also be reached in parallel, directly over the Internet, with more media than just voice.

1.2 Connecting to the outside world

Within the European NREN community, openness is key, both in sharing knowledge and in establishing electronic relationships. Therefore, N-ECS supports not only SIP trunking, but also an open connectivity model. Since this openness may also impose potential threats, such as Spam for Internet Telephony (SPIT), the task force that assembled N-ECS has paid attention to developments that enable you to set up open but secure channels by using Transport Layer Security (TLS) encryption.

1.3 Call scenarios

This handbook deals a lot with call scenarios. In general, it is good to keep the following basic structure in mind when call scenarios are discussed:



In all cases we refer to the initiator of a session, whatever media are being used, as the 'calling party'. The person that the calling party intends to reach is the 'called party'. This basic scenario is illustrated regardless of the technologies used along the way. Any combination of technologies can be used, adding intermediate components for transport, conversion and added functionality.

1.4 General terms

There are many protocols involved in the various functions of VoIP services. However, the two most popular protocols used for the signalling of calls are H.323 and SIP. Interoperability of clients and services relies primarily on signalling and secondarily on the exchange of media. For a general discussion of VoIP, check http://en.wikipedia.org/wiki/Voice_over_IP and the old *IP Telephony Cookbook*.

1.4.1 VoIP codecs

The term codec refers to the method of coding/decoding media data (voice and video) to meet bandwidth requirements of the available network channel. A caller device has to support the same codecs as those available on the other side, i.e., the called device or server. Otherwise the exchange of media may not take place, even though call signalling negotiation has been successful. There are many codecs available for voice, the most popular of which are the G.xxx series as standardised by the International Telecommunications Union (ITU). The most commonly available codecs are G.711 - Pulse code modulation (PCM), demanding a 64 kbit/s channel, and G.729 - Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP) demanding an 8 kbit/s channel. For a general discussion of voice codecs, see http://en.wikipedia.org/wiki/Speech_encoding.

1.4.2 VoIP services

Attempting a simple classification of VoIP services, for the purposes of this handbook, we see:

- basic services: registration of endpoint on servers, call routing, location and call translation;
- value-added: voice-mail, missed call notification;
- enhanced communication services: interoperability of voice calls with other media, like video, instant messaging, as well as trusted domain infrastructure to support level of trust indication to the end user.

1.4.3 Terminal - Endpoint - Agent - Client - Phones

All terms refer to the device (phone) or software (soft-phone) that the end user utilises to place and receive calls. Phones are termed ‘endpoints’ in H.323 terminology, ‘agents’ in SIP terminology and all other terms are used interchangeably.

1.4.4 Gatekeeper – Registrar - Redirect - Location-Proxy Server

Phones in most cases require a server to register to, before receiving calls, so they can be located while mobile. This requirement, as well as the need for reliable caller identification, imposes the need to authenticate phones before allowing them to connect to their local domain server. These servers are termed ‘gatekeepers’ in H.323 terminology and ‘registrar-redirect-location-proxy’ servers in SIP terminology.

1.4.5 Gateway

A gateway is a device that allows calls between two devices which are using incompatible protocols or interfaces and can not directly call each other. For example, a SIP phone needs a SIP <=> H.323 gateway in order to call an H.323 endpoint. Also, a SIP phone needs a SIP <=> PSTN gateway in order to call a traditional Private Branch Exchange (PBX) or Public Switched Telephone Network (PSTN) phone line. A gateway converts the signals between the two end devices and terminates media channels on both sides, acting as intermediary for the call.

1.4.6 Media proxy

A media proxy is an intermediary device that terminates media channels, but does not handle signalling at all. Media proxies are used mostly for traversing firewalls, bypassing Network Address Translation (NAT), or protecting domains from direct access by outsiders, by allowing calls to reach the boundary of a domain without opening up the whole network.

1.4.7 PBX

A Private Branch Exchange is a traditional telephone centre, used in most organisations or enterprises, responsible for connecting (switching) telephone ports between subscribers (local phones). Recently, some PBXs have been replaced by IP-PBXs which rely on VoIP at their core, rather than switched circuitry.

1.4.8 Call routing

This is the functionality of VoIP servers to route incoming calls to their destination. This task is easy for calls placed between two users on the same local server. However, in most cases, an incoming call may indicate a destination (dialled number) on a remote server and in a format that needs interpretation. In this case the routing protocols that the local server implements will need to check and translate the incoming dialled number. Usually, part of the number consists of a 'prefix' or 'domain' that points to the next hop (server or telephony switch) in the infrastructure.

Call routing consists of multiple sub-tasks:

- locating the far endpoint - selecting the destination endpoint from the list of possible destinations, according to local routing policy. SIP location is mostly determined based on Domain Name System (DNS). H.323 location services can use DNS based location too, but using DNS is not common practice yet. If the Uniform Resource Identifier (URI) is an E.164 number, then the location service can use Telephone Number Mapping (ENUM) to resolve the E.164 phone number to a SIP URI. The domain can be extracted from the URI and the domain can then be resolved by looking up service records (SRV) for that domain in order to get the destination IP address.
- URI manipulation - if the URI is a 'TEL' URI (in the form TEL:<number>), or if the user part contains a number then usually the number needs to be normalised. Non-numerical characters, such as dashes and brackets, should be removed and the number should be completed as agreed upon by the receiving party. Good practice is to always send a number in the form of <Country Code><Area Code><Subscriber Number><Extension>, without preceding zeros and without a zero between the country code and the area code (some countries form an exception to this). So 31308008008 is the full E.164 telephone number for extension 08 of subscriber number 80080 within Utrecht (030) in the Netherlands (31).
- forwarding of signalling packets to the selected server.

1.4.9 NAT traversal

Phones that are located behind NAT networks are usually at a disadvantage, as communication with their local server is obstructed by the NAT. There are many methods to help traverse the NAT barrier, which vary based on the VoIP signalling protocol. These methods require support on the local server, on the phone itself and sometimes they require the existence of a Media Proxy as well. Additional protocols have been developed to augment the traversal of NAT, like ALG, STUN, ICE, TURN, and H.460 in the case of H.323. Refer to the old *IP Telephony Cookbook* for an explanation of these terms.

1.5 SIP-related terms

1.5.1 Domain

The Session Initiation Protocol supports the use of domains. This can be compared to the area code in a telephone number, e.g., 020 represents Amsterdam in the Netherlands. With SIP the domain can be alphanumeric, just like domains used in e-mail addresses. An address has the same form as an e-mail address: user@domain.org. An advantage is that users will have one single address where they can be reached both by e-mail as well as by IP telephony and videoconferencing, and it is easier to remember than a telephone number.

1.5.2 DNS-based routing

Again, just like e-mail, the domain of a user address can be used to find the server that the user is registered with. If user_a@domain_a.org wants to reach user_b@domain_b.org, either the client of User A or the N-ECS of User A can do a DNS look-up for an SRV record that resolves domain_b.org to the DNS name of the N-ECS of User B, which could be sip.domain_b.org for instance. See the request for comment RFC 3263 (http://www.ietf.org/rfc/rfc3263.txt).

1.5.3 Identity

In general, when the word 'identity' is used in this handbook, it refers to the electronic profile of the user as it is stored in a database on the N-ECS. It at least contains a username, a password, a SIP address and telephone number. In the authentication phase the user must prove to the N-ECS that he is the person that this profile relates to, by providing unique credentials such as a username/password. In a broader sense, this identity can reside in a master database within the user's home institution. Following the single sign on principle, a user should be able to authenticate using their institutional credentials. These credentials should be checked in a secure channel through an Authentication and Authorisation Infrastructure (AAI).

1.5.4 TLS

Transport Layer Security is a well-known mechanism to identify a communicating party on an IP connection, and to encrypt the communication channel. It is commonly used for securing the communication between web browsers and web servers. Regularly, an icon in the form of a closed lock indicates the connection is encrypted, and the identity of a server is verified by examining its electronic certificate which can be compared to a passport in the physical world. The certificate, usually based on the X.509 standard, is handed out by organisations that have proved trustworthy of handling identity information. When a browsing user also needs to present a certificate, the authentication is referred to as being mutual. This is usually the case with servers that communicate with each other. SIP servers can transport the SIP messages over an encrypted channel as well. User clients should check the validity of the server certificate before providing credentials. Both credentials and SIP messages are not readable when intercepted by hackers. Communication between SIP servers uses mutual authentication.

1.5.5 SIP.edu

SIP.edu is an association of higher education institutions in the USA, which work together on Internet-based technologies. It supports the SIP.edu working group (http://www.internet2.edu/sip.edu/) that has described optimal ways of installing, configuring and deploying SIP-based communication infrastructures within academic institutions. Their 'recipes' are deployed at a number of institutions, resulting in a significant number of users reachable through SIP.

1.5.6 ITAD/ISN

Until the entire world can be reached through SIP addresses in the same form as e-mail addresses, telephone numbers will remain in existence. Also, since it is cumbersome to enter an e-mail address on a (cell) phone or Personal Digital Assistant (PDA), the Internet Telephony Administrative Domain (ITAD) Subscriber Number (ISN) was constructed by Internet2 and the Massachussets Institute of Technology (MIT). It resolves numbers formatted like 1234*256 to any URI, for example to a SIP URI such as user@domain.org. The first part identifies a user in an ITAD. The second part, after the '*', is the domain part (ITAD) of the number.

The advantage of ITAD is that it does not use E.164 numbers at all. A disadvantage is that the numbers are not portable, in contrast with E.164 and ENUM. ITAD uses a hybrid solution, an ENUM private tree named 'freenum.org', like {*itad*}.freenum.org. So if a user's ID is 1234 and the ITAD is 256, the ITAD number can be resolved to Number Authority Pointer (NAPTR) record 4.3.2.1.256.freenum.org. Further resolution is handled the same way as with ENUM. The ITAD register is maintained by the Internet Assigned Numbers Authority (IANA). See also http://freenum.org/cookbook/

1.6 Telephony terms

1.6.1 E.164

Defined by the International Telecommunication Union (ITU), all telephone numbers in the world follow certain regulations. The way telephone numbers are constructed, starting with a country code (see http://en.wikipedia.org/wiki/List_of_country_calling_codes), followed by an area code or service prefix and then the subscriber number, is defined in the E.164 standard (see http://en.wikipedia.org/wiki/E.164). This also refers to country code definitions, number lengths and so on. This ensures that not only end users know exactly how to reach others, but also telephony network operators know how to interconnect with other operators.

1.6.2 ENUM

Telephone Number Mapping or E.164 Number Mapping is a mechanism to resolve a URI or IP address from an E.164 number, using a telephone number as a starting point and getting a contact URI information. Any query containing an official E.164 telephone number can be sent to the DNS infrastructure, and if the party that officially holds this number has entered any contact information, DNS will return for instance a SIP address, H.323 address, e-mail address or many more information items. This information can be used by the calling party to decide what technology to use to reach the called party. Agreements about entering and validating the information in DNS is part of the ENUM framework and may differ from country to country. In each country, one organisation is responsible for delegating the entries for national numbers within reach of the country code. This official ENUM tree is also referred to as the 'golden tree' that uses e164.arpa as the DNS root. Other DNS trees are in use that do not use the e164.arpa root. All of those are called private trees despite the fact that most can be queried publicly, like E164.org and NRENum.net. Delegation procedures for those trees and their roots differ. Also, DNS trees exist that can only be queried by selected parties. If these trees mainly contain routing information between (voice) carriers and can only be queried by selected parties, this application of ENUM technology is called 'infrastructure ENUM'.

1.7 Other abbreviations

Further specific abbreviations that are used in this document are described below:

FQDN	Full Qualified Domain Name
Prio	Priority
TTL	Time To Live

2 Supported scenarios

Before setting up the N-ECS, it is important to determine what role it will play, since it has many features. After that, it is explained how it can be downloaded, configured and tested.
Many software packages are available, both commercial and open source, that in some or many ways offer real time communication features. Not one product can provide all features, such as providing presence, making voice conversations over the Internet possible, adding the option of making phone calls to regular phones, adding Private Automatic Branch Exchange (PABX) functions, adding videoconferencing and so on. Therefore, some initiatives combine various products to provide as complete as possible a set of features; for example a SER/OpenSER/Asterisk-based solution is provided by the Internet2 SIP.edu initiative.

What N-ECS does is to combine the three most commonly used open source server products in one distribution. Not just one, like a fancy version of Asterisk, but the three most commonly used server products within the European academic community. On top of that, it provides examples for integrating them as well. Even more, it adds state of the art features, especially in the area of security. It thus supports the functionality necessary for the following roles:

- Institutions or schools (or companies) can use it as:
- experimentation server for testing
    - VoIP (both SIP and H.323)
    - videoconferencing
    - instant messaging and presence
    - secure communications
    - ENUM;
- local gatekeeper within the Global Dialling Scheme (GDS);
- an example of a PABX replacement with many value-added services like ENUM, presence and videoconferencing.

- Internet Service Providers (ISPs) or NRENs can use it also for
- Base National Gatekeeper install

The N-ECS is not intended to be used in a production environment. If you consider using some of its functionality in a production environment, the given N-ECS implementation can be used as a complete test environment for developing production-like services.

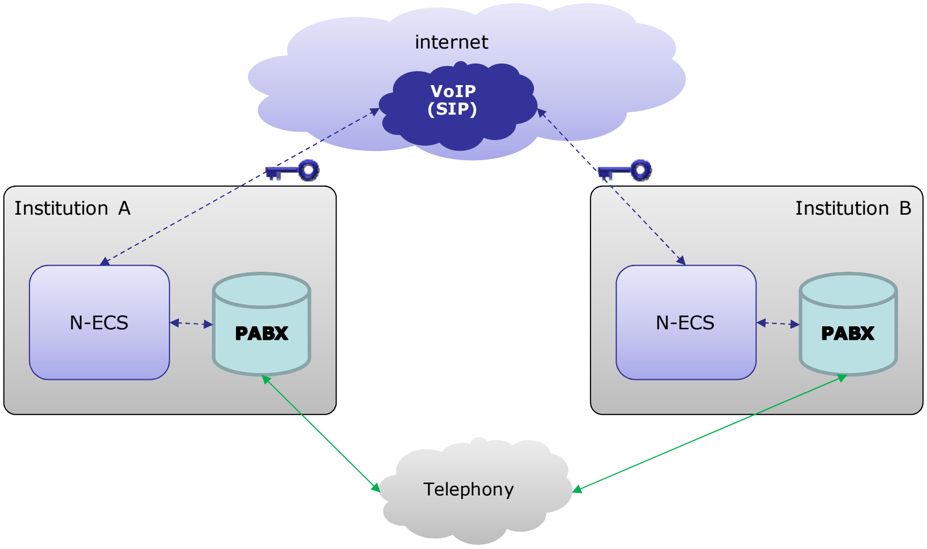
2.1 Making the island accessible

The added value of N-ECS lies in securing communication with the outside world. There are a couple of ways that communication can cross organisational borders:

2.1.1 Inter-institutional communication

Within the international academic community, bandwidth is usually no obstacle and openness is in the very nature of the organisations. Considering these observations, it would make sense if academic institutions exchanged their telephony traffic over Internet-based technology. In some countries, this is actually the case, but it is not as common as might be expected.

The Global Dialling Scheme is an example of an overlay infrastructure based on H.323 that enables institutions to easily find and reach each other for phone- and videoconferences. The possibility for SIP to find the called party easily through DNS is also slowly gaining ground. However, SIP-based (commercial) PABXs are very likely not set up to allow SIP calls from other institutions over the incoming Internet connection. The most widely used connection protocol between VoIP implementations is still ISDN (in Europe) or equivalent. N-ECS can act as an intermediary between the untrustworthy Internet and the PABX. It is not relevant whether the PABX is a traditional Time Division Multiplexing (TDM)-based PABX or a SIP-based PABX. In the first case, the N-ECS will connect to the PABX through ISDN or any other available TDM interface. In the second case, a SIP trunk between the two can be established. This trunk does not have to be secured since it will not be visible to the outside world. Most vendors that offer VoIP-based PABXs offer SIP trunking as a feature, though it might involve expensive 'gateways' that convert their specific VoIP flavour to SIP. In all cases, N-ECS allows for incoming sessions over the Internet in a secure fashion, as shown in the following diagram:



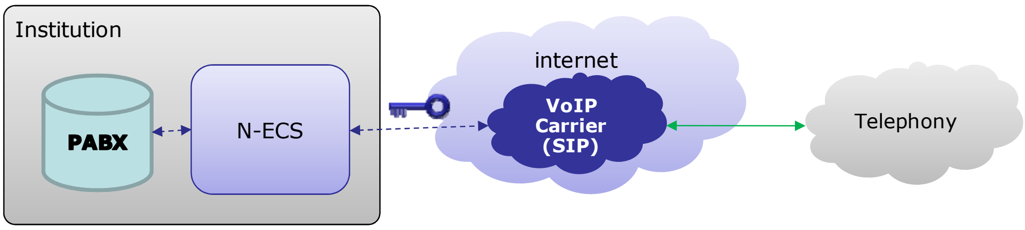
The main reason for this scenario can be cost savings, but the business case is very dependent on how much intra-institutional traffic is exchanged. Another argument for using this is to broaden the range of possible applications that can be made available to end users. Current infrastructures focus on voice only, while SIP enables a much wider spectrum of services which support the end user in their day to day electronic communication. In the scenario as it is described, the combination of N-ECS and PABX together make the 'home server' in terms of call scenarios, unless the N-ECS is actually functioning as the PABX itself.

Asterisk is known to be able to act as a full-grown PABX by itself and was designed as such in the first place. For a deployment of Asterisk as a full PABX, please refer to the Asterisk home page on how to set it up so that it meets production environment needs such as redundant set up.

2.1.2 Institution-carrier interconnect

An upcoming market is that of carriers which offer SIP-based VoIP connectivity as an alternative to ISDN. In general, a Virtual Private Network (VPN) or dedicated connection point is established for this purpose. Depending on the business model, it is most suitable for an institution to use the existing high bandwidth connection. The same features as offered by ISDN services are provided, such as number screening, online billing and so on, but the connection scales more easily in terms of capacity as long as there is enough bandwidth available. Adding more capacity is mainly a matter of the carrier allowing more concurrent sessions, which might affect the fee for the connection. Some carriers offer free-of-charge calling to their customers and even fewer support ENUM.

If the institutional PABX supports SIP trunking, the relationship with the VoIP carrier might be established directly. If the PABX is of the traditional kind, or VoIP based but compatibility problems occur, N-ECS can play a role as the intermediate component. In a diagram, this scenario is shown as follows:



In the diagram, it is assumed again that the VoIP-based PABX allows for SIP trunking, but the N-ECS provides additional security. It also adds features that the VoIP-based PABX has not implemented (yet), such as ENUM support.

If the PABX is TDM based, the N-ECS is definitely necessary as an intermediate with the addition of an ISDN gateway, to make the connection between the PABX and the VoIP carrier. Encryption between the institution and the VoIP carrier can be achieved by a VPN. The VPN endpoints need to have excellent real-time performance like high throughput, low latency and jitter and no packet loss. A better option is to encrypt the SIP channel using TLS, precisely one of the unique features of the N-ECS. Support for TLS is not common yet amongst VoIP carriers, unfortunately.

All scenarios mentioned above can be combined, with N-ECS playing different roles at the same time. For more in-depth information regarding interconnect scenarios, please refer to the *IP Telephony Cookbook*.

2.1.3 Carrier-carrier interconnect

Among carriers, the use of VoIP has been common for quite a while. ENUM as a technology (not the procedures that are part of the ENUM definition) is used to determine where to route calls among them. This type of interconnect is out of the scope of this document, but is worthwhile mentioning to show that the technologies used here are the same as those used by high-volume, high-performance commercial companies.

Another reason for mentioning this application of VoIP is that VoIP is part of a large portion of the long-distance telephone conversations that you make, without being aware of it. On the one hand it shows that VoIP will not much endanger voice quality, on the other it means that the voice information has been

transformed from and to VoIP one or more times. Each time, buffers add to the delay, echo cancellers are needed and codec conversion might degrade the voice quality eventually.
It is worthwhile to keep the voice path free of these occurrences, which pleads for starting and ending using VoIP without any intermediate conversions.

3 Before you begin

Before installing N-ECS, the following points are important to go through. Important information regarding the versioning of the components is followed by an explanation of how to obtain the N-ECS. An overview of the available features rounds up this chapter, after which you are prepared to set up your own N-ECS.

3.1 Versioning

The versions of the main components, that are currently part of the distribution of the NREN-Enhanced Communications Server, are:

- OpenSER 1.3.2-tls;
- Asterisk 1:1.4.21.2 with common H.323 module;
- GNUgk 2.2.7;
- Debian 4 release 2 – lenny testing version. The testing version was chosen because most of the packages for the applications above are lagging behind the stable version, and N-ECS is intended to demonstrate the latest release. An OpenSER build was installed from an external repository because the Debian package of OpenSER omits TLS in order to avoid OpenSSL licensing issues.
- The VMware image as described in this document runs on both Windows and Linux versions of VMware Server.

3.2 How to get the latest version

The latest version of N-ECS can be found at: http://www.terena.org/n-ecs/.
The N-ECS is not intended to be used in a production environment. If you consider using some of its functionality in a production environment, the given N-ECS implementation can be used as a complete test environment for developing production-like services.
Note that many issues need to be addressed when putting a server such as N-ECS into production, like hardening (i.e., closing all necessary ports, minimising root access to binaries, not permitting direct root login through SSH, partitioning based on user rights and so on).

3.3 Prerequisites

To test the NREN-Enhanced Communication Server, you need at least:

- a physical server - usually, a PC with a network card will suffice if you do not want to route hundreds of calls per minute.
- a connection to an internal network using a private IP address, or a connection to a publicly accessible network using a public IP address. If clients connect from other locations, the server should be connected to a publicly accessible network. Setting up the server within a private network and letting clients connect from other locations using NAT to reach the server is out of scope of this document and for some functionality even impossible.
- a host name.
- full firewall protection for the network during the first configuration steps, except for access to web sites that contain updates, in order not to become a victim of abusers of vulnerabilities. Beware that network interfaces are not enabled by default in VMware and that the firewall is prepared to run just after changing the IP address.

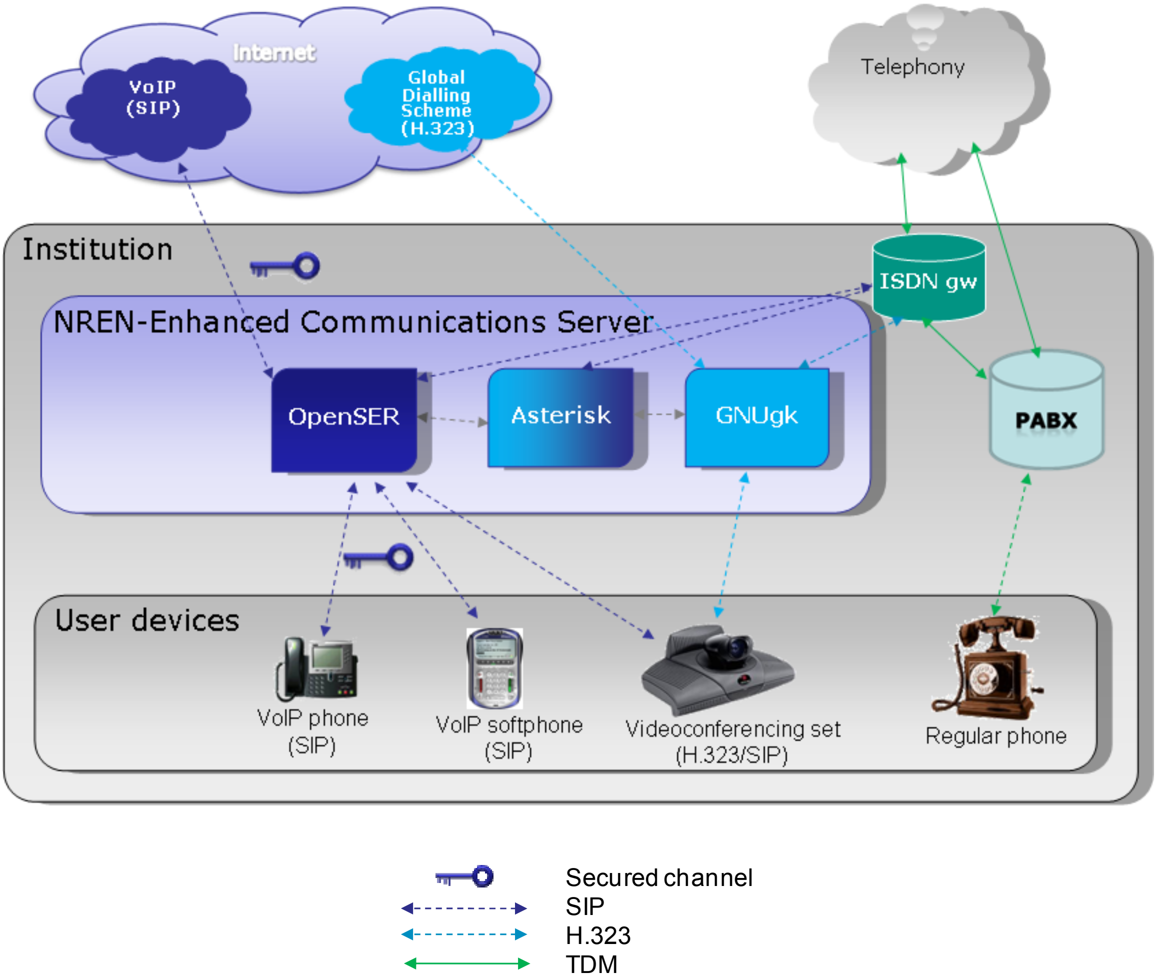
3.4 Supported features

The features supported by N-ECS are basically those of all components that were used, combined. If you intend to use only the features of one of the components, it is worthwhile to look at other packaged products that specialise in that product, like Trixbox as a Graphical User Interface (GUI) -based version based on Asterisk for instance. The value of N-ECS lies in examples of how to combine Asterisk, GNUgk and OpenSER, and in how to secure communication to and from the server - currently a relatively new and hot topic in the area of real-time Internet communications.

The most important supported features are:

- voice calls
- video calls
- instant messaging
- Presence
- Echo server
- SIP peering
- secured SIP over TLS for client-server communication and for server-server communication
- supported protocols: both H.323 and SIP are supported
- supported infrastructures: Global Dialling Scheme (H.323), SIP.edu, ENUM, NRENum.net

When N-ECS is deployed within an institution, the relationship between its components and the outside world is depicted in the diagram below. The three main components of N-ECS are drawn within one container, and the user devices, or clients, connect one or more of the components.



4 Recipe for setting up the NREN-Enhanced Communications Server

After you have gone through the steps in the previous chapter, you are ready to set up the N-ECS.

4.1 Step 1: Prepare your server and download and install N-ECS

N-ECS can be run directly on a physical machine, or virtualised for testing purposes. To ensure real-time performance and to support Plain Old Telephone Service (POTS) / ISDN Peripheral Component Interconnect (PCI) cards, it is strongly recommended to run N-ECS directly on a physical machine.

4.1.1 Prepare the physical server

N-ECS is configured to run on Debian 4.0r2 testing. Follow these steps to get the physical server up and running:

- Set up your physical server by either booting from a network install CD or any other Linux live CD or DVD. See http://www.debian.org/CD/netinst/ for the Debian netinst download location.
- Download and boot the Debian netinst or Linux live image.
- Partition and format your destination disk.
- Mount your new destination hard drive to /mnt.
- cd /mnt
- Download the N-ECS packages from the server.
  Find the most recent version on http://www.terena.org/n-ecs/ and use wget to download the .tar.gz file.
- tar –xvzf /home/yourname/n-ecs_<version>.tar.gz
- Make sure that the boot loader (GRUB in this case) is (re)installed in the Master Boot Record (MBR) of your main disk. See https://help.ubuntu.com/community/RecoveringUbuntuAfterInstallingWindows for additional help.

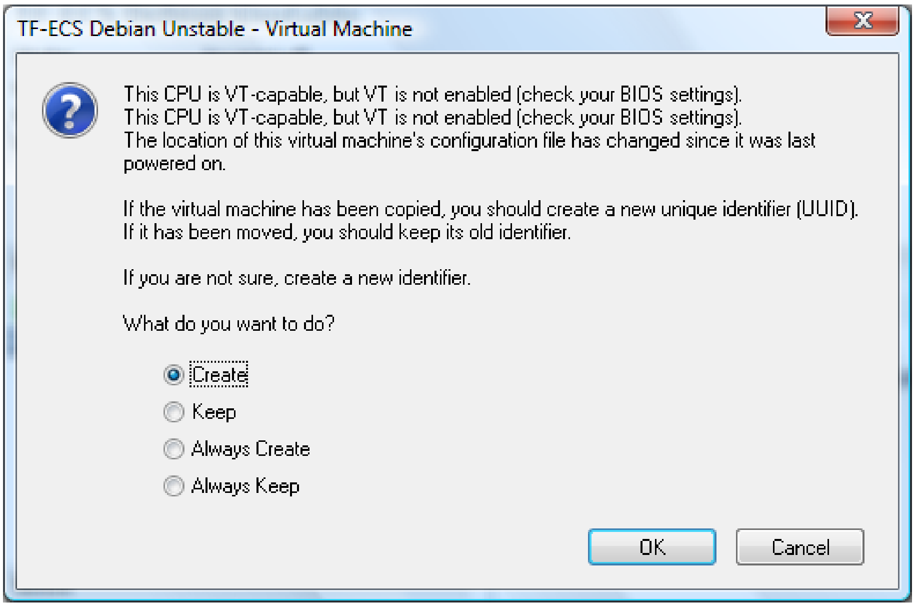
You can skip the following paragraph and continue setting up the basic configuration.

4.1.2 When using the VMware image: Install VMware

Using a virtualised N-ECS gives you freedom in which operating system to use on the physical machine. Unless hundreds of clients generate many concurrent sessions, the N-ECS will not consume many resources, so for testing purposes it is convenient to use an existing test server that does not need the operating system to be reinstalled. The disadvantage of using a virtualised N-ECS is that the real time performance cannot be guaranteed. In the worst case this might result in intermittent loss of sound and the system clock of the virtual machine not running in sync with the system clock of the physical host.

Depending on the choice of your operating system, the instructions for installing and accessing VMware will differ. This document describes how VMware runs on top of Microsoft Windows.

- When (re)booting your physical host machine, enter the Basic Input Output System (BIOS) and enable the feature that allows virtualisation support for your processor. The way to enter the BIOS and the location of the setting differs for every vendor, so consult the vendor's documentation to find the appropriate setting.
- Download the VMware server version that is most suitable for your host operating system from http://www.vmware.com.
- Download the N-ECS VMware image from http://www.terena.org/n-ecs. Assume it will be downloaded to c:\download\.
- Unpack the VMware image by using Winzip, 7zip or any utility that can handle zipped tarball files (.tgz), i.e. to c:\virtual machines\.
- Install VMware according to the installation instructions for your host operating system.
- Start VMware by clicking the VMware console.
- Open the preconfigured virtual machine by choosing File -> Open and choose c:\virtual machines\N-ECS\n-ecs.vmx.
  The virtual machine is configured to use 512 MBytes of internal memory, which is enough for up to 20-25 test users.
- In the VMware console, click 'start this virtual machine' to activate the virtual machine.
- You will get the following message, stating that it is advisable to create a new unique identity for the virtual machine, since it was created on a different host:



TF-ECS Debian Unstable - Virtual Machine

This CPU is VT-capable, but VT is not enabled (check your BIOS settings).
This CPU is VT-capable, but VT is not enabled (check your BIOS settings).
The location of this virtual machine's configuration file has changed since it was last powered on.

If the virtual machine has been copied, you should create a new unique identifier (UUID).
If it has been moved, you should keep its old identifier.

If you are not sure, create a new identifier.

What do you want to do?

● Create
○ Keep
○ Always Create
○ Always Keep

  - choose 'create'.
- A warning that the CPU is VT-capable, but VT is not enabled, tells you to enter the BIOS of the host machine and enable this feature (see the first of these steps).

Do not forget to press ctrl+alt if you want to leave the virtual machine window within the VMware console and you want to get your mouse cursor back on the host operating system.

4.2 Step 2: Change the basic configuration

For the first configuration steps it is wise to work from the console instead of through secured shell (SSH). The Ethernet interface of the virtual machine is not connected during booting up, to prevent any conflicts and attacks during the configuration phase. Root access via SSH is disabled from outside by default.

- Login as user 'root' with password 'n-ecs'.
- Change the password by typing:
  `passwd`
- Create a new user using the command:
  `adduser <yourusername>`
- Use your favourite editor (vi, vim, nano, joe) to set up the IP address, mask, gateway and DNS servers by editing */etc/network/interfaces*.
  If you want to use Dynamic Host Configuration Protocol (DHCP), ensure that the mapping is made permanent in your DHCP server settings.
- Set the hostname by editing /etc/hostname and type:
  `hostname –F /etc/hostname`
- Edit */etc/hosts* and replace the first entry with your IP address, full host name and short host name.
- Edit */etc/iptables.conf* and update the IP address.
- Bring the network interface down and up again by typing:
  `/sbin/ifdown eth0`
  and
  `/sbin/ifup eth0`
- Connect the Ethernet interface in the VMware console and set it as 'connected at power on', for example by clicking on the interface card icon in the lower right-hand corner of the window.
- Test the connectivity by using the 'ping' command to the gateway IP address.
- Update ssh by typing:
  `/apt-get update sshd`
  and restart the daemon:
  `/etc/init.d/sshd restart`

From now on, you can log onto the (virtual) server using SSH. The configuration will not allow you to log on directly as root. This could be changed in */etc/ssh/sshd.config* by changing the 'PermitRootLoginValue'.
Now you should change the IP address in configuration files. It is set to A.B.C.D everywhere where it needs to be changed. You may also use the command:

```
        find /etc | xargs grep A.B.C.D
```

to find all places where it occurs.

4.2.1 Change the Asterisk configuration

Edit the files */etc/asterisk/h323.conf* and */etc/asterisk/sip.conf* to change the IP address ('bindaddr' parameter value) to your host IP address, and in *h323.conf* set the host name of the gatekeeper to be the same as the host name of your server.

Restart Asterisk by typing:

```
        /etc/init.d/asterisk restart
```

4.2.2 Change the OpenSER configuration

Edit the file */etc/openser/openser.cfg* and replace all (6) occurrences of A.B.C.D with your virtual machine's IP address. For example, change the following values:

```
Listen=       udp:<IP address of N-ECS>:5060
Listen=       tcp:<IP address of N-ECS>:5060
Listen=       tls:<IP address of N-ECS>:5061
```

Change the Alias parameter to the name of your domain:

```
Alias=<your SIP domain>
```

If you want to have TLS fully functional (so that the remote client is able to verify the server certificate) you should also adjust certificate settings. You can find out to do this in Section 4.4 'Configure and test TLS'.

Restart OpenSER by typing on the Linux command line:

```
/etc/init.d/openser restart
```

If you change anything in the configuration file, you have to restart the server. OpenSER doesn't have a reload command like GnuGK or Asterisk.

You can monitor OpenSER by using:

```
/usr/sbin/openserctl monitor
```

Or view the logging in

```
tail -f /var/log/openser.log
```

You might want to set the debug level higher temporarily in */etc/openser/openser.cfg*.

We choose to use the Least Cost Routing (LCR) module as the primary repository for local number routing. Although routing can be done directly in the configuration script, there is a need to restart the server after such a change. This is not necessary with LCR. So you can add a route to Asterisk by typing on the command prompt:

```
/usr/sbin/openserctl lcr addgw_grp local 1
/usr/sbin/openserctl lcr addgw GnuGK <IP address of your server> 5070 1 1 1
'' 0
/usr/sbin/openserctl lcr addroute 95000114 '' 1 1
/usr/sbin/openserctl lcr addroute 95000115 '' 1 1
```

This routes prefixes 95000114 and 95000115 to Asterisk over User Datagram Protocol (UDP). Asterisk version 1.4 is not able to receive SIP over Transmission Control Protocol (TCP), so if the routing is done in the script you have to use the 'UDP' parameter within the 't_relay' function.

Configuration changes executed using openserctl, such as adding LCR routes, do not require the restarting of OpenSER. The command is:

```
openserctl lcr reload
```

Unfortunately, not all modules allow you to make these changes in this way. Furthermore if you restart OpenSER this configuration will be lost, so it is wise to store as much of this information as possible in a

database. Be aware though that, if you change database parameters directly, e.g. using mysql commands, you have to instruct the server to load the information from the database.
You can easily see what routes are configured by using:

```
/usr/sbin/opensercctl lcr show
```

4.2.3 Prepare DNS

The minimum required DNS configuration consists of SIP SRV records in DNS. Since many types of DNS servers exist, the way to enter the configuration for all of them is out of scope of this document. Be sure to enter the following DNS entries in the configuration for <yourdomain>:

Record	Type	Prio	TTL	xxx	Port	Reference
_sip._udp	SRV	0*	3600*	100	5060	<FQDN of your N-ECS>
_sip._tcp	SRV	0*	3600*	100	5060	<FQDN of your N-ECS>
_sips._tcp	SRV	0*	3600*	100	5061	<FQDN of your N-ECS>
_h323ls._udp	SRV	0*	3600*	100	1719	<FQDN of your N-ECS>
<FQDN of your N-ECS>	A	0*	3600*			<IP address of your N-ECS>

* any value that is suitable in your case

Optionally, you can also set up NAPTR records for your domain. These records return sets of services for the domain together with accompanying preferences. By doing this you have more control over setting up the preferred transport types for SIP. Conversely, SRV records do not enforce the type of transport to use for any order; the resolver determines beforehand what type of transport to use and queries the corresponding SRV record, so the service provider cannot determine the transport preference. This can be ensured by NAPTR records. The following example shows how to set up SIP services with a preference for transport in the order: TLS first, then TCP, and finally UDP. These records can enforce the use of TLS (even without using the SIP secure (SIPS) scheme) and provide an alternative to TCP and UDP in case of problems.

Record		Type	DER	PREF	flags	SERVICE	REG EXP	REPLACEMENT
<your domain>	IN	NAPTR	100	0	“s”	“SIPS+D2T”	“”	_sips._tcp.<yourdomain>.
<your domain>	IN	NAPTR	200	0	“s”	“SIP+D2T”	“”	_sip._tcp.<yourdomain>.
<your domain>	IN	NAPTR	300	0	“s”	“SIP+D2U”	“”	_sip._udp.<yourdomain>.

You can always check the NAPTR records using the command:

```
host -t naptr yourdomain
```

4.2.4 Set up the firewall

The N-ECS uses the host-based IPTables firewall. Additionally, you might want to add a network firewall, or you will have to adjust an existing network firewall. Depending on the services you want to provide, the following ports are used by the N-ECS and are set up in the */etc/iptables.conf* file:

Port	Type	Protocol	Service
22	TCP	SSH	OpenSSH
5060	UDP	SIP	OpenSER SIP
5060	TCP	SIP	OpenSER SIP
5061	TCP	TLS	OpenSER SIP
10000-19999	UDP	RTP	Asterisk Media traffic (Audio)
1718	UDP	H.323	GnuGK
1720	TCP	H.323	GnuGK
20000-20999	TCP	H.323 Q.931	GnuGK
30000-30999	TCP	H.323 H.245	GnuGK

4.2.5 Understand the numbering plan

By default, a numbering plan is configured in the N-ECS that assigns number blocks to the various functions of the N-ECS:

- 950001100 - 950001109: unused
- 950001110 - 950001130: SIP extensions configured in OpenSER
- 950001140 - 950001159: Asterisk services
  - 950001140: echo test
  - 950001141: echo test without announcements
  - 950001142: static voicemail
  - 950001143: static voicemail
- 950001160 - 950001190: H.323 extensions configured in GnuGK

The country code is configured as 999 by default. You can adjust it in */etc/openser/openser.cfg* and */etc/gatekeeper.ini.*

Routing of blocks related to protocols is set in GnuGK and Asterisk. Routing of Asterisk blocks has to be setup in OpenSER according to Section 4.2. Routing of H.323 blocks has to be set up in OpenSER according to Section 4.4. OpenSER uses numbers only as aliases. Primary identifiers are names, according to sip.edu. This allows you to add more users than you have numbers available in your numbering plan, although users without number aliases will not be reachable from H.323.

4.2.6 ENUM

ENUM is active by default on your N-ECS. For every call set up, first the official ('golden') ARPA tree and then the NREN tree (NRENum.net) is queried to resolve the number to either a SIP or H.323 address. If you want to change the ENUM functionality, see Section 4.6.2 'Expand ENUM'.

Many countries have an official entry for their country code, but the national branch is not filled. Queries to those ENUM branches consume time without any rewriting rules as results. Even this could cause a delay in call set up that you should be aware of.

4.3 Step 3: Set up extensions

4.3.1 Create SIP test accounts

Initially we will test SIP accounts that reside in OpenSER. To do so, follow these steps:

- Log on to the server through an SSH client.
- Create two users, for instance *user1* and *user2*, using the following command:
  ```
  /usr/sbin/openserctl add <username> <password> <email>
  ```
- Add a numeric alias for the users, using the following commands on the commandline:
  ```
  /usr/sbin/openserctl aliasdb add user1@<your domain> 950001111@<your
  domain>
  /usr/sbin/openserctl aliasdb add user2@<your domain> 950001112@<your
  domain>
  ```

Using the 'openserctl alias db' command you can also set aliases for services like the echo service:

-
  ```
  /usr/sbin/openserctl aliasdb add echo@<your domain> 950001140@<your
  domain>
  ```

Registered endpoints can be seen using the command:

-
  ```
  /usr/sbin/openserctl ul show
  ```

4.3.2 Install and configure a test SIP client for the first users

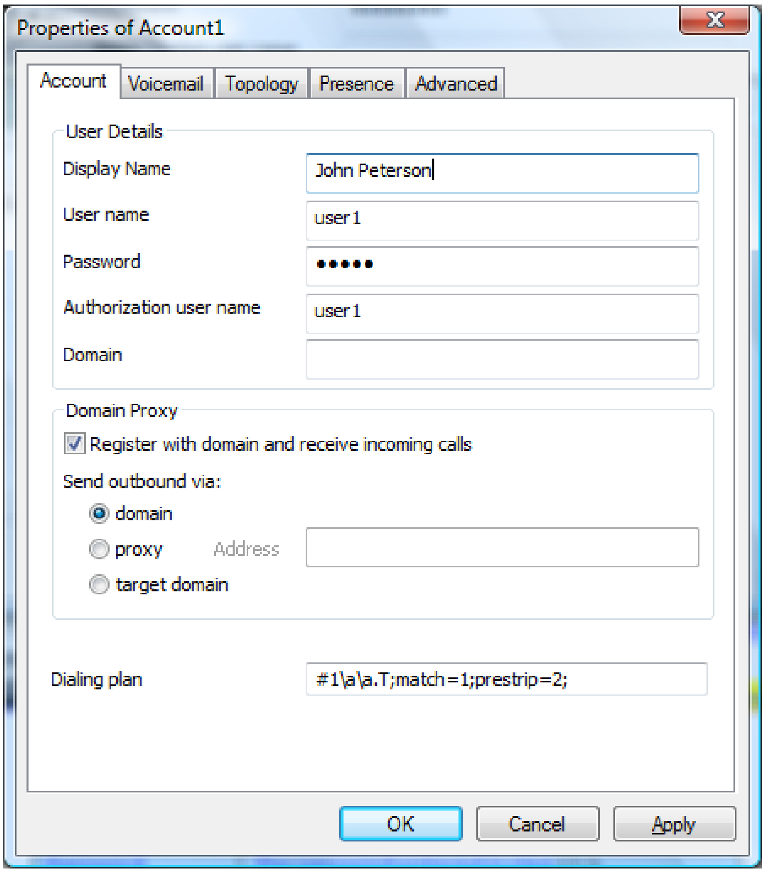
For initial testing, there are a couple of freely available SIP clients. For Windows, the clients that were tested are:

- X-Lite,
- SJPhone
- Ekiga
- Grandstream GXP2000 (SIP phone, no TLS)
- ALLnet AL7960 (SIP phone, no TLS)
- FRITZ!Box FON WLAN 7170 (SIP, no TLS)
- CounterPath EyeBeam and Bria

Ekiga is a good candidate if you test clients on Linux.

The initial test can be done using X-Lite, which supports video but does not support the use of TLS to secure communications between the client and the server. If you need to test using TLS right away, there are commercially available SIP clients, such as Counterpaths Bria. SJphone also supports H.323, which can be useful later on. SJphone supports audio only. The aim of this document is not to provide a complete overview of available SIP clients and a comparison of their features; they are used for demonstration purposes only.

- Download X-Lite from http://www.counterpath.com/x-lite.html.
- Run X-Lite.
- Right click on the X-Lite window, and choose 'SIP Account Settings'.
- Choose 'Add...' and fill in the new window as below:



Properties of Account1

User Details

Display Name: John Peterson

User name: user1

Password: •••••

Authorization user name: user1

Domain:

Domain Proxy

☑ Register with domain and receive incoming calls

Send outbound via:

● domain

○ proxy Address

○ target domain

Dialing plan: #1\a\a.T;match=1;prestrip=2;

Be sure to replace 'yourdomain' with the actual name of your domain.

- The values of the fields should be filled in according to this table:

Field	Value
Display Name	choose freely, *e.g., John Peterson*
Username	the username entered in Step 4.3.1, *e.g., user1*
Password	the password entered in Step 0
Authorisation Name	<can be left blank>
Domain	the SIP domain (as set in alias parameter in openser.cfg)

- Click 'Close'.

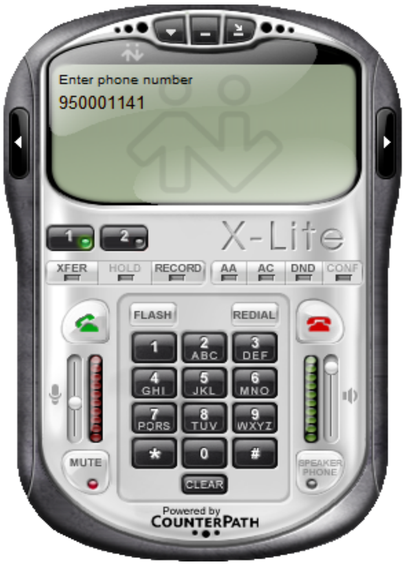
If you have valid DNS SRV records (see 4.2.3) then filling in the 'domain' field is sufficient. Otherwise, put the IP address or Fully Qualified Domain Name (FQDN) of your SIP server in the 'proxy' field.

4.3.3 Make a test call

Initially, you can make a test call to the echo service, which sends back the audio that you send to the server through your client. This enables you to observe the audio quality and the delay between sending and receiving the audio. The advantage of this test is also that you need only one client.

4.3.3.1 Make an echo test call

In X-Lite, type '950001141' and press enter (or click the green phone icon on the middle left-hand side of the X-Lite window).



Enter phone number
950001141

4.3.3.2 Make a client-to-client test call

Make sure at least two users are logged on to the N-ECS server, or that you know someone on another server who is reachable for a test call.

- Assuming you are logged on as user1, type the other user's address, e.g., 'user2@yourdomain', or 'user2' or the short number 12.
- X-Lite should make a ringing sound and the other user should get an incoming call.

4.4 Step 4: Prepare the server for H.323

Another special feature of N-ECS is the combination of H.323 support and SIP support. With a small adjustment, the GnuGK server on the N-ECS can be configured so that it accepts registrations from H.323 endpoints and transfers calls to the SIP domain. Further on, you will find information about how to connect the GnuGK to your national gatekeeper in case you are interested in support for the Global Dialling Scheme.

Edit the file */etc/gatekeeper.ini* and change the IP address under the following section:

```
[RasSrv::PermanentEndpoints]
<IP address of your N-
ECS>:2020=asterisk;95000111,95000112,95000113,95000114,95000115
```

Restart the GnuGK by typing:

```
/etc/init.d/gnugk restart
```

Furthermore, you have to let OpenSER know how the GnuGK can be reached by typing on the Linux command line:

```
/usr/sbin/openserctl lcr addroute 95000116 '' 1 1
/usr/sbin/openserctl lcr addroute 95000117 '' 1 1
/usr/sbin/openserctl lcr addroute 95000118 '' 1 1
/usr/sbin/openserctl lcr addroute 95000119 '' 1 1
```

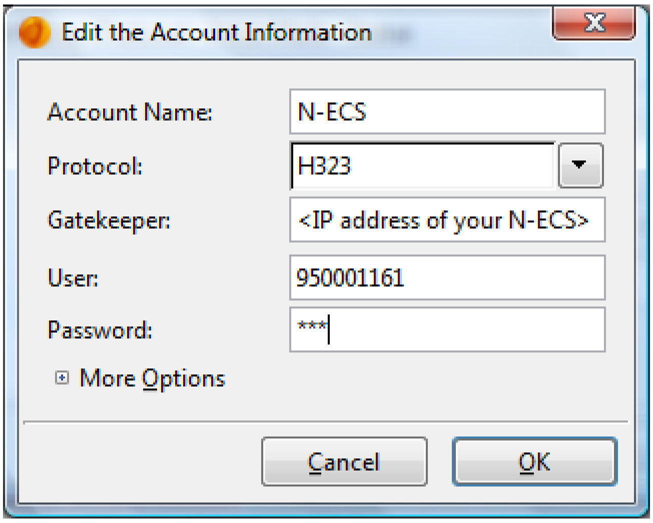
Asterisk is configured to route SIP and H.323 number blocks according to its configuration file:

/etc/asterisk/extensions.conf

4.4.1 Install and configure a test H.323 client for first users

SJPhone supports H.323 in addition to SIP. It only supports audio. If you want to test video, or test H.323-to-SIP calls and previously had SJPhone configured, it is worthwhile to use Ekiga in this test.

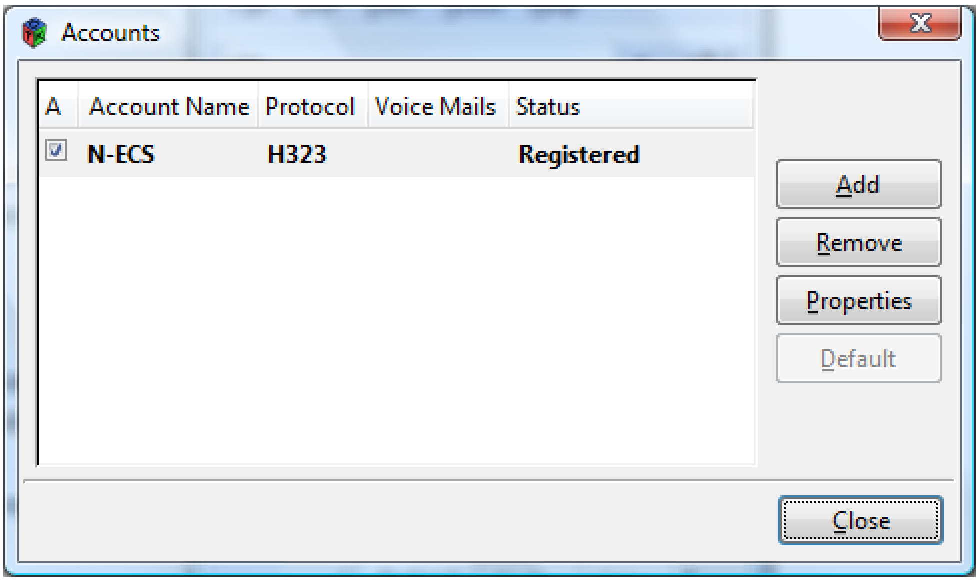
- Download Ekiga from http://www.ekiga.org/.
- Install Ekiga, ignoring the Ekiga account set up.
- In the menu of the main window, choose 'edit' -> 'accounts' and add an account.
- Configure the new account as shown below:



Account Name:	N-ECS
Protocol:	H323
Gatekeeper:	<IP address of your N-ECS>
User:	950001161
Password:	***

You can use any password; the current GnuGK configuration accepts open registrations.

- In the account overview, activate the account by clicking the check box to the left of its name:



A	Account Name	Protocol	Voice Mails	Status
☑	N-ECS	H323		Registered

In the 'Status' column the registration status will be shown, which should be 'Registered'.

- Make a test call to 'h323:950001141', which will lead you to Asterisk's echo service.

Check *'Tools'* -> *'General History'* to see the logging in Ekiga in case problems occur.
On the server, you can get more information in GnuGKs logfile:

```
tail -f /var/log/gnugk/gnugk.log
```

or

```
less /var/log/gnugk/gnugk.log
```

Note : See registered endpoint http://<IP address of N-ECS>/gk/vdir.php . You can also see them through the console (telnet localhost 7000) by typing the command '?' or 'rv'.
If you want to observe current calls on the GnuGK, open a web browser that points to:
http://<IP address of N-ECS>/gk/vcalls.php.

4.5 Step 5: Configure and test TLS

The current certificate that is used for SIP-over-TLS is derived from a self-signed Certification Authority (CA). Such a certificate cannot be verified easily or almost-automatically by a remote party.

In order for sessions from your N-ECS to be trusted by other parties (SIP servers and SIP clients), one approach is to exchange all the operational N-ECS servers' root CA certificates amongst administrators of those infrastructures. This requires a lot of coordination and does not scale. For production purposes, it is recommended to obtain a certificate that is more widely trusted, that can be provided by:

- your institution's CA
- TERENA's CA: http://www.terena.org/activities/scs/
- a commercial CA

This way, you will need to spend a lot less effort to connect to other domains. If the certificate is from one of the publicly available CAs it is not necessary to exchange server certificates before users can reach each other, but it is still necessary to add them to the OpenSER 'ca_list' file that contains the list of authorities to be trusted by OpenSER. This can be done by simply copying the CA certificate into the file and restarting the server.

NOTE: the downside of using a commercial CA and adding it to the list is that all certificates issued by this CA will be trusted and you cannot control who receives these certificates. Additional checks may be needed, or additional policies may have to be set by the CA, to build trusted domains for real-time communication. NREN CAs can provide a more flexible approach to this.

Note: another approach would be to use one CA for all certificates of N-ECS servers. However, this would greatly reduce the freedom to choose your own CA later on. Using one CA from which all server certificates are derived also imposes the problem of vendor lock-in and is practically impossible.

For testing purposes it is easy and affordable to use a self-signed certificate. OpenSER provides a simple way to get one. First you need to decide whether to create your own CA or use the one provided with the server. If you do not want to create a CA, skip the 'Create a CA' step.

4.5.1 Create a CA

- Move to */etc/openser/tls*.
- Back up the content of the directory and delete the root CA and user directory.
- Adjust the file 'ca.conf' everywhere you find a '#please update' at the end of the line.
- Run the command:
  `openserctl tls rootCA`
  and set your CA password.

4.5.2 Create a server certificate

- Move to */etc/openser/tls*.
- Copy 'server.conf' to a new file '<yourservername>.conf'.
- Adjust '<yourservername>.conf' everywhere you find a '#please update' at the end of the line. You should at least update commonName and subjectaltName (server name and SIP domain name).
- Run the command:
  `openserctl tls userCERT <yourservername>`
  and use your CA password (which is 'n-ecs' in case of the supplied CA).

You can find your server certificate, key and CA list in the '<yourservername>' directory.

Adjust the following lines in */etc/openser/openser.cfg* to match your file names:

```
tls_certificate = "/etc/openser/tls/yourservername/yourservername-cert.pem"
tls_private_key = "/etc/openser/tls/yourservername/yourservername-
privkey.pem"
tls_ca_list = "/etc/openser/tls/yourservername/yourservername-calist.pem"
```

In order for the changes to take effect, restart OpenSER by typing:

```
/etc/init.d/openser restart
```

4.5.3 Install and configure a test SIP client for first users using TLS

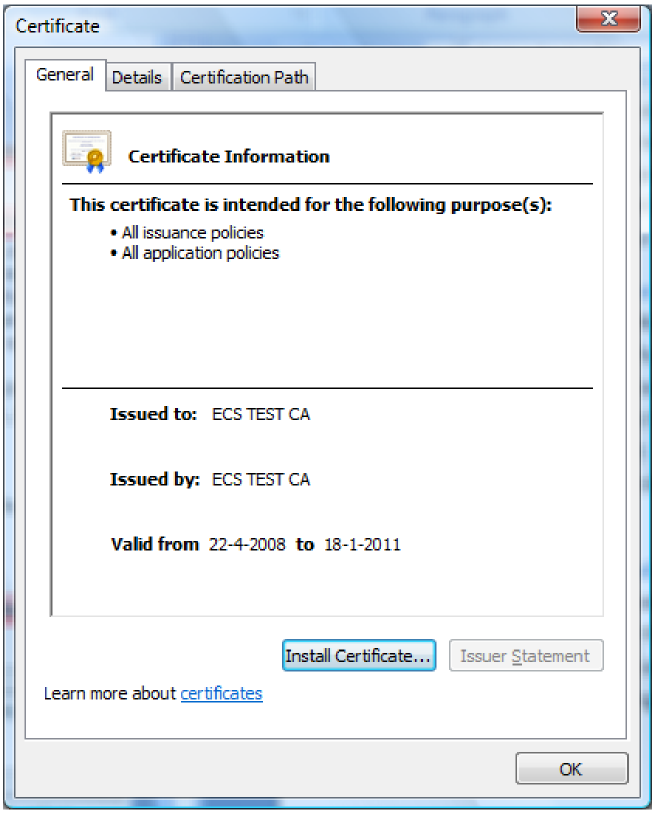
As said earlier, with some endpoints you can test secured communications with your N-ECS server. Therefore you need to trust the server's root X.509 certificate. Later, you can request a commercially available certificate for your N-ECS, so that most computers will trust your N-ECS automatically, since a number of (commercial) root certificates are installed by default, for instance on Windows, and are available as a package for Linux.

4.5.3.1 Installing the N-ECS server root certificate on the client computer

- Download the N-ECS security root certificate from your server or the commercial CA.
- Store the certificate with the extension 'crt' in a local folder.

Installation of the root certificate depends on your operating system and version. The following steps describe how to install it with Microsoft Windows XP and Vista:

- Open necs-ca.crt



Certificate Information

This certificate is intended for the following purpose(s):

- All issuance policies
- All application policies

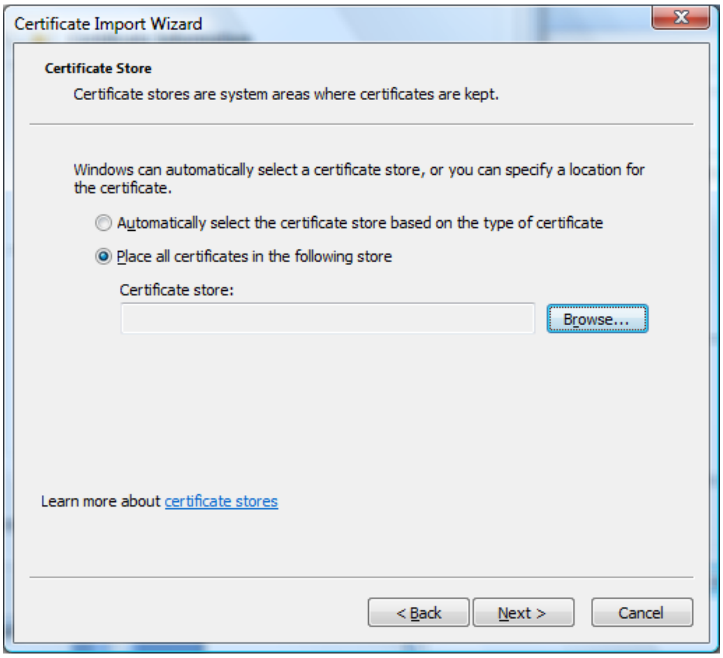
Issued to: ECS TEST CA

Issued by: ECS TEST CA

Valid from 22-4-2008 **to** 18-1-2011

Learn more about certificates

- Click 'install certificate' and then click on the 'welcome' screen of the Certificate Import Wizard.
- In the next window, choose 'Place all certificates in the following store'.


Certificate Import Wizard

Certificate Store

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

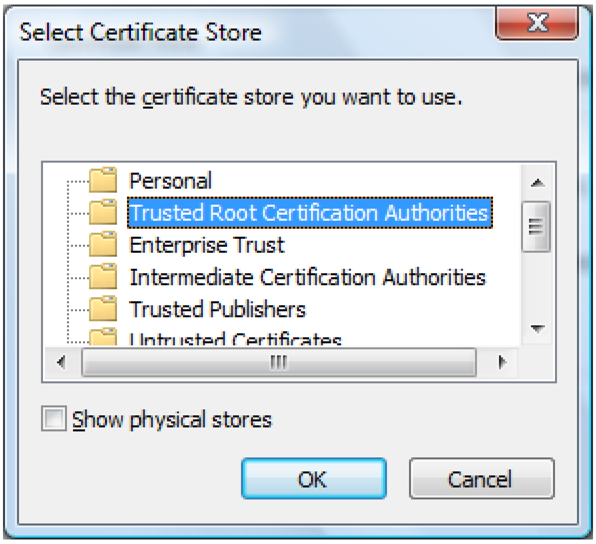
○ Automatically select the certificate store based on the type of certificate

● Place all certificates in the following store

Certificate store:

Learn more about certificate stores

- Click 'Browse' and choose the 'Trusted Root Certification Authorities' store.


Select Certificate Store

Select the certificate store you want to use.

Personal
Trusted Root Certification Authorities
Enterprise Trust
Intermediate Certification Authorities
Trusted Publishers
Untrusted Certificates

□ Show physical stores

- Click 'OK', 'Next' and 'Finish'. The root certificate is installed.

4.5.3.2 Installing and configuring a TLS compatible client

At the time of writing, no working version of a freely available SIP client was found to support SIP over TLS. Some commercial clients, however, already support SIP over TLS. If you intend to use Bria or Eyebeam, the commercial SIP client of Counterpath, your configuration will be similar to that of X-Lite. Additionally, under the 'Security' tab of the SIP account data, you should change the protocol to 'TLS'. If you set up NAPTR records for your domain according to Section 4.2.3, the automatic transport protocol set up by the NAPTR record should occur.

4.6 Step 6: Set up peering based on TLS

Exchanging sessions, or simply peering, is easily achieved with the use of SRV records. If you have entered the appropriate records, other parties can find your N-ECS and therefore also the users who are connected to your N-ECS or the extensions that are connected to a PABX that is served by the N-ECS. If these sessions come in over UDP or TCP it is not easy to reliably verify their origin. By using a blacklist, unwanted servers can be blocked, based on incoming domains and IP addresses, but this can only take effect after the first damage has been done, i.e. users were harassed with unwanted calls. By (only) allowing TLS, you can at least verify the origin of the calling party's home server and decide whether or not to trust calls coming from this server.
The need for an acceptable certificate is obvious in this case, so check Section 4.4 on the requirements, how to obtain one and how to install it. The certificate should be trusted by the servers of other domains that you want to reach. If this is achieved by either exchanging the root CA certificates of the servers, or by installing public certificates, your server is ready to peer with others.
You can simply copy (using the 'cat' command) another CA or server certificate of your federation partner to *yourservername-calist.pem* file to ensure it is trusted by your server.

When the TERENA task force TF-ECS started configuring the N-ECS, it was meant to support SIP-Identity. This promising technology combines the use of TLS with a verifiable identity of the calling party, so that an incoming session can be trusted based on the identification of the actual calling party and not of their home server. At the time of writing, the technology is not yet mature enough to be provided by the N-ECS. It is advised to follow any developments in this direction and stay aware of safe ways of peering with other parties.

4.6.1 Change to your national locale

In order to connect to the Global Dialling Scheme for H.323 calling and videoconferencing, your N-ECS should be registered with the National Gatekeeper of your country. To find the contact information for the service provider (usually your NREN), please refer to http://www.terena.org/activities/tf-ecs/gds.html.
You will receive information about the IP address that you will forward calls to and accept calls from, and what number block will be routed to you. You need to enter this information in the following section in the */etc/gatekeeper.ini* configuration file of GnuGK on your N-ECS:

```
[Neighbor::national]
Host=<FQDN or IP address of the National Gatekeeper in your country>
AcceptPrefixes=*
SendPrefixes=*,!<prefixes supported by your N-ECS that should not be sent to
the National Gatekeeper>
ForwardResponse=1
ForwardHopCount=10
```

You can test the connection to the GDS by calling someone you know who is connected to the GDS, or by registering another endpoint to a GDS gatekeeper that anyone can connect to. Such gatekeepers are called Free Love Gatekeepers. More information can be found on http://contact.surfnet.nl/freelove/.

To let OpenSER deal with the country code and prefix (which consists of the area code and a large part of the local number, except for the extension), find the following lines and adjust them to your needs:

```
## local number
if (uri=~"^sip:(999)?9500011[0-9][0-9]@") {
        if (uri=~"^sip:999") {strip(3);}
}
if (uri=~"^sip:[1-9][0-9]@") {
        prefix("9500011");
}
```

If your country code consists of two digits, replace 'strip(3)' with 'strip(2)' and so on.
If you change the local prefix (9500011), be sure that you also change the settings in Asterisk (in the files *extensions.conf*, *sip.conf* and *h323.Conf* located in */etc/asterisk/*) and GnuGK.

4.6.2 Expand ENUM

As said before, querying ENUM is active by default on your N-ECS. You can choose to query trees other than the 'golden tree' and the NRENum.net experimental tree that are configured by default.

To let GnuGK query additional trees, add to */etc/gatekeeper.ini* i.e.:

```
ENUMservers=e164.arpa,nrenum.net,e164.org,
```

To let OpenSER query other ENUM trees, replace the tree root in the following line or copy and edit the block in */etc/openser/openser.cfg*:

```
if ( !enum_query("nrenum.net") )
```

Also, you might find it useful to enter the numbers supported by your N-ECS in ENUM, so that others can query them to be able to use the most efficient technology that is at hand.

In order to enter the information in the 'golden tree', find an ENUM Registrar that can support your numbers or number blocks. The availability and procedures differ significantly per country. Please refer to the following URLs for more details:
http://enumdata.org/
http://www.ripe.net/enum/

If you only intend to experiment with ENUM and are a member of an academic institution, you can request to register your number(block)s with the NRENUM.net tree. Find out who is the contact person for your country at http://nrenum.net/index.php/Delegation_Request.

4.7 Future features

The TERENA Task Force on Enhanced Communications Services ended its activities in October 2008, but its members are still active in the areas described here. Features that TF-ECS determined to be useful and that you can also contribute to are:

- voicemail;
- call forwarding;
- IPv6 support;
- monitoring of the N-ECS and using the N-ECS as a monitoring node for end points or other N-ECS nodes;
- directory integration for OpenSER and GnuGK;
- Asterisk 1.6rc2 with TLS support;
- combined dial plan (single number for multiple protocols).

5 Reference to European implementations

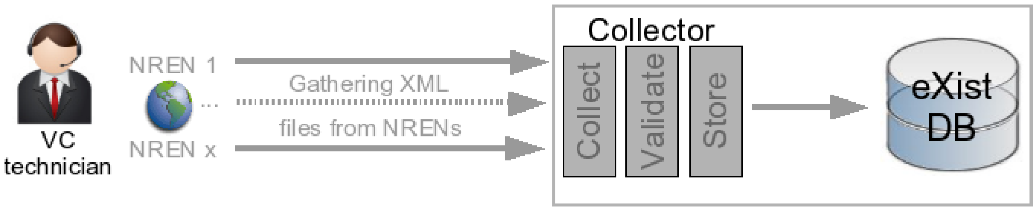
SA6 is a Service Activity within the GN2 project framework, started in year 4 of the project with the agreement of a number of NRENs providing advanced national VoIP and IP videoconferencing services. The service umbrella created and maintained by GN2 SA6 is publicly referred to as 'educonf'.

The primary goal of educonf is to create a European higher-education and research videoconference and VoIP coordination service by supporting service administrators and end users with helper services, such as:

- a ticketing system for coordinated resolution of international VoIP- and videoconference-related reachability problems;
- a federated directory of VoIP and videoconference service deployments to support international cooperation and the establishment of new partnerships between research and education players;
- a distributed monitoring service to record national VoIP and videoconference connectivity in order to improve problem identification and resolution;
- a knowledge base to support NREN administrator and end user problem solving;
- a Multi-point Control Unit (MCU) videoconferencing resource in support of European research projects and for NRENs that currently do not have a multi-point videoconferencing infrastructure, thereby bridging the digital divide between parts of Europe and between international higher education and research projects.

The educonf VoIP/VC database is a federated directory that collects information on NREN and institutional videoconference and VoIP deployments from locally maintained data sources. The principle idea is to create an XML based data schema that will convey local organisational VoIP or videoconference information to a central server, that enables querying by end users and administrators relying on data sources provided by data owners themselves (using a pull model). The original concept was developed in TERENA TF-ECS, but the scope of the information collected has been significantly broadened.

The figure below shows the basic architecture of the educonf directory service. The directory entry (a completed XML file) is stored at the particular organisation and maintained by local VoIP / videoconference administrators. These directory entries are regularly collected, validated and stored by a central service component.



A directory entry can describe the following service details:

- basic organisational data, contact points, locations;
- prefixes belonging to the particular organisation and the dialling schemes they are connected to (call routing methods);
- gatekeepers and SIP proxies serving the zones / prefixes / domains, use policies for third parties;
- free gatekeepers;
- available MCU services, MCU gateway services and use policies for third parties;
- availability of videoconference venues or endpoints and their reachability, contact points and locations;
- test numbers for equipment connectivity and quality testing;
- VoIP / videoconference service contact points.

At the time of writing a pilot educonf directory service is available. As a starting point, visit: http://educonf.geant2.net/services/directory/.