

# VOIP

lien :

[Qu'est-ce qu'un téléphone VoIP ? | Zoom](#)

[Setting Up a VoIP Server with Asterisk on Debian 12 | Reintech media](#)

[La VoIP et la ToIP pour les nuls - FRAMEIP.COM](#)

[Asterisk : Mettre en place un menu interactif \(IVR\) avec Google TTS - DSI et chef de projet IT externalisé sur Nantes.](#)

[Comment vérifier son adresse IP sur Linux: 12 étapes](#)

[https://github.com/thierry-rami/Asterisk\\_config/blob/main/Voip.pdf](https://github.com/thierry-rami/Asterisk_config/blob/main/Voip.pdf)

**Q**uel est le coût opérationnel et de maintenance d'un service **VOIP** ?

**L**a VoIP, les coûts opérationnels et de maintenance peuvent également être importants et doivent être pris en compte lors de la planification et du déploiement d'un système VoIP, qu'il soit basé sur Asterisk ou sur toute autre plateforme VoIP.

**V**oici quelques éléments à considérer en ce qui concerne les coûts opérationnels et de maintenance spécifiques à la VoIP :

1. **Connectivité Internet** : Les appels VoIP nécessitent une connexion Internet stable et fiable. Vous devez prendre en compte les coûts associés à la bande passante Internet nécessaire pour prendre en charge vos appels VoIP, ainsi que les éventuels frais mensuels de fournisseurs de services Internet.
2. **Frais de licence ou d'abonnement** : Certains systèmes VoIP commerciaux peuvent impliquer des frais de licence ou d'abonnement, en fonction des fonctionnalités et de l'échelle du déploiement.
3. **Coût des équipements** : Outre le logiciel VoIP lui-même, vous devez tenir compte des coûts des équipements physiques tels que les téléphones IP, les passerelles VoIP et autres appareils nécessaires pour votre infrastructure VoIP.
4. **Frais de configuration et d'installation** : L'installation initiale d'un système VoIP peut nécessiter des frais de configuration et d'installation, en particulier si vous engagez des consultants externes pour le faire.
5. **Mises à jour et maintenance logicielles** : Comme mentionné précédemment, les mises à jour logicielles régulières sont nécessaires pour maintenir la sécurité et les performances de votre système VoIP. Cela peut entraîner des coûts de maintenance liés au temps et aux ressources nécessaires pour appliquer ces mises à jour.
6. **Support technique** : Si vous choisissez d'utiliser un système VoIP commercial, vous pouvez également avoir besoin de payer pour un support technique supplémentaire pour obtenir de l'aide en cas de problèmes ou de difficultés.

**E**n conclusion, bien que la VoIP puisse offrir des économies significatives par rapport aux systèmes téléphoniques traditionnels, il est important de prendre en compte les coûts opérationnels et de maintenance associés à la mise en œuvre et à la gestion d'un système VoIP.

**C**es coûts peuvent varier en fonction de la taille, de la complexité et des exigences spécifiques de votre déploiement VoIP.

---

**P**ourriez-vous dire en quoi la configuration VoIP d'un call center serait différente de la configuration **VoIP** d'un standard téléphonique d'une entreprise ?

**O**ui, la configuration **VoIP** d'un centre d'appels (call center) peut être différente de celle d'un standard téléphonique d'une entreprise en raison des besoins spécifiques et des exigences opérationnelles des centres d'appels.

**V**oici quelques différences clés entre les deux :

**1. Volume d'appels et échelle :**

- Un centre d'appels traite généralement un volume beaucoup plus élevé d'appels entrants et sortants par rapport à un standard téléphonique d'entreprise. Par conséquent, la configuration VoIP d'un centre d'appels doit être dimensionnée pour gérer ce volume élevé de trafic.

**2. Fonctionnalités avancées de routage des appels :**

- Les centres d'appels nécessitent souvent des fonctionnalités avancées de routage des appels pour acheminer efficacement les appels vers les agents appropriés. Cela peut inclure le routage basé sur des critères tels que la disponibilité des agents, les compétences requises, la priorité de l'appel, etc.

**3. Gestion des files d'attente :**

- Les centres d'appels utilisent généralement des files d'attente pour gérer les appels entrants en attente d'être pris en charge par un agent disponible. La configuration VoIP doit prendre en charge la gestion efficace de ces files d'attente, y compris les temps d'attente, les annonces d'attente, la musique d'attente, etc.

**4. Intégration CRM et outils de centre d'appels :**

- Les centres d'appels peuvent nécessiter une intégration étroite avec des systèmes CRM (Customer Relationship Management) ou d'autres outils de centre d'appels pour fournir des informations contextuelles aux agents et pour enregistrer les données d'appel pertinentes.

**5. Surveillance et reporting avancés :**

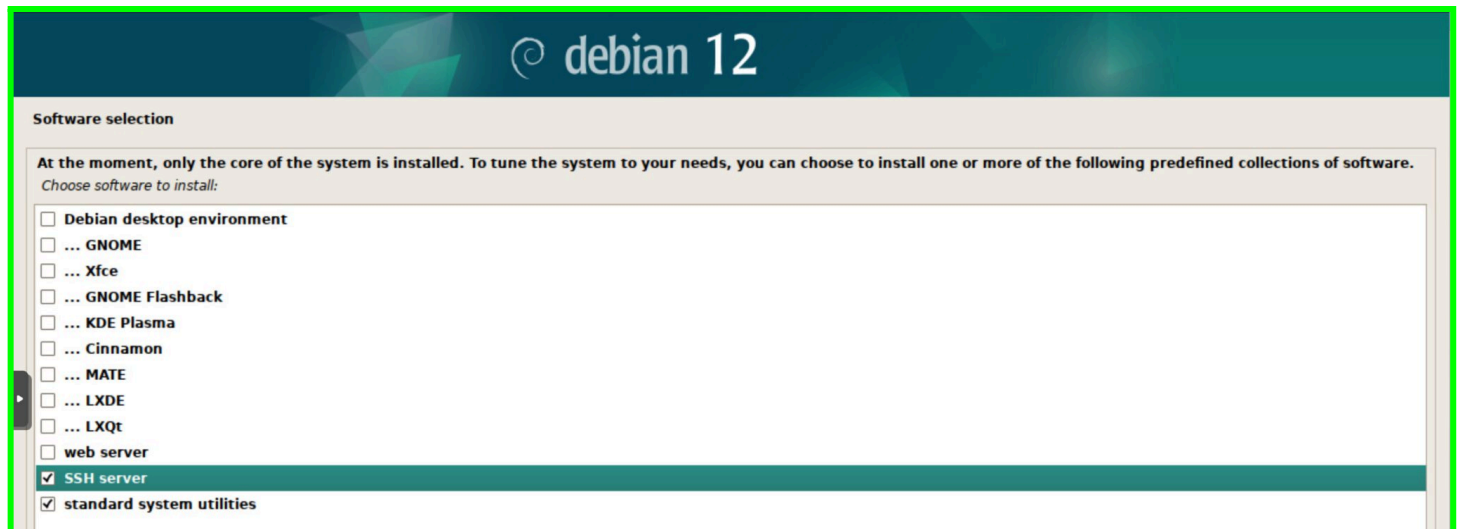
- Les centres d'appels ont souvent besoin de capacités avancées de surveillance et de reporting pour suivre les performances des agents, le temps d'attente des appels, les taux de réponse, etc. La configuration VoIP doit permettre la collecte et l'analyse de ces données de manière efficace.

**6. Flexibilité et évolutivité :**

- Les centres d'appels peuvent avoir des besoins de flexibilité et d'évolutivité plus importants en raison de variations saisonnières ou de campagnes spécifiques. La configuration VoIP doit être conçue pour être flexible et évolutive afin de répondre à ces exigences.

**E**n résumé, la configuration VoIP d'un centre d'appels diffère souvent de celle d'un standard téléphonique d'entreprise en raison des volumes plus élevés d'appels, des fonctionnalités avancées de routage et de gestion des appels, de l'intégration avec des outils de centre d'appels, des besoins de surveillance et de reporting avancés, ainsi que de la nécessité de flexibilité et d'évolutivité.

## 1) Debian installation



Attention ! ne pas oublier d 'installer les paquets **SUDO**

```
sudo apt install sudo OU sudo apt get install sudo
```

```
laurent@debian:/etc/asterisk$ sudo apt install sudo
[sudo] Mot de passe de laurent :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
sudo est déjà la version la plus récente (1.9.13p3-1+deb12u1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
laurent@debian:/etc/asterisk$ _
```

## 2) Preparation environnement

2.1 Après avoir installé Debian, nous avons déjà des outils utiles tels que nano,sudo, wget et tar dans le package. Nous installerons également Curl pour nos besoins et mettrons à jour le système d'exploitation.

```
1: apt update
```

```
2: apt upgrade
```

```
3: apt install curl
```

2.2 Nous désinstallerons le système de contrôle d'accès Apparmor par défaut. Cela pourrait interférer avec le fonctionnement de notre Astérisque:

```
1: systemctl stop apparmor
```

```
2: apt remove apparmor
```

2.3 Nous allons maintenant télécharger le programme d'installation d'Asterisk et le décompresser.

```
1: cd /usr/src
```

```
2: wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20-current.tar.gz
```

```
3 :tar zxvf asterisk-20-current.tar.gz
```

```
4: rm -rf asterisk-20-current.tar.gz
```

Dans la prochaine étape, nous préparerons l'environnement pour compiler et exécuter le serveur Asterisk sur le système Debian.

Nous utiliserons les outils et le script `install_prereq` fournis avec le projet Asterisk, qui permettent d'automatiser le processus d'installation.

Le script `install_prereq` identifie et installe les dépendances système nécessaires pour compiler et exécuter Asterisk, ainsi que les bibliothèques, outils ou autres composants nécessaires au bon fonctionnement d'Asterisk.

Si vous souhaitez savoir exactement ce que fait ce script, cela vaut toujours la peine de consulter la documentation d'Asterisk ou de lire les commentaires dans le script `install_prereq` lui-même, qui peut contenir des informations sur ce sujet.

```
1 : cd asterisk-20*/
```

```
2 : contrib/scripts/install_prereq install
```

### 3) Asterisk Compilation

La commande suivante lancera le processus de configuration du logiciel, qui est requis avant la compilation proprement dite.

Les fichiers de configuration utilisés pendant le processus de construction seront générés.

Ces fichiers contiennent des informations sur la configuration de votre système et les options de build.

Il s'agira de Makefiles ou de scripts de construction utilisés pour compiler le logiciel.

Si, par habitude, nous exécutons le `./configure` commande avec les paramètres affichés dans cet avertissement, nous ne pourrons pas exécuter Asterisk.

```
1: ./configure --with-srtp
```

La commande `--with-srtp` permet de rajouter dans le make menu les modules SRTP pour les sélectionner après!

```
1: make
```

Selon les options que vous choisissez, la compilation peut prendre un certain temps.

À la fin, nous recevrons des informations satisfaisantes similaires à celle ci-dessous:

```
+----- Asterisk Build Complete -----+
+ Asterisk has successfully been built, and +
+ can be installed by running:             +
+                                           +
+               make install               +
+-----+-----+-----+-----+-----+
```

## 4) Installation du logiciel ASTERISK

La prochaine commande **make install** installera le logiciel compilé sur le système.

```
1: make install
```

Après une courte installation, nous recevons le message suivant:

```
+---- Asterisk Installation Complete -----+
+      YOU MUST READ THE SECURITY DOCUMENT  +
+ Asterisk has successfully been installed. +
+ If you would like to install the sample  +
+ configuration files (overwriting any     +
+ existing config files), run:             +
+                                           +
+ For generic reference documentation:      +
+     make samples                         +
+                                           +
+ For a sample basic PBX:                  +
+     make basic-pbx                      +
+                                           +
+----- or -----+
+ You can go ahead and install the asterisk +
+ program documentation now or later run:   +
+                                           +
+               make progdocs              +
+                                           +
+ **Note** This requires that you have     +
+ doxygen installed on your local system   +
+-----+-----+-----+-----+-----+
```

Nous pouvons maintenant, comme suggéré, générer des exemples de fichiers de configuration et les déplacer vers le répertoire des exemples, où nous pourrions les utiliser comme base pour une configuration ultérieure de nos services.

```
1: make samples
```

```
2: mkdir /etc/asterisk/samples
```

```
3: mv /etc/asterisk/*.*/etc/asterisk/samples/
```

La commande suivante créera une configuration et un scénario de base **PBX** (Private Branch Exchange) dans **Asterisk**.

Le scénario PBX est un exemple de configuration d'Asterisk comme central téléphonique privé permettant d'établir des appels entre des numéros de téléphone internes.

Voici ce que fait exactement la commande **make basic-pbx**

- 1 : **Créer une configuration de base** : La commande génère une configuration Asterisk de base, qui comprend des fichiers de configuration pour les utilisateurs, les extensions, les contextes et les routes sortantes.
- 2: **Configuration utilisateur** : La commande peut créer des configurations utilisateur, ce qui vous permet d'établir des connexions entre eux au sein du système PBX.
- 3: **Configuration des extensions** : le scénario PBX contient des exemples d'extensions qui définissent les actions à entreprendre lorsque les utilisateurs appellent des extensions spécifiques.
- 4: **Contextes** : La configuration du PBX est divisée en contextes qui définissent quels numéros de téléphone et extensions sont disponibles dans quelle partie du système.
- 5: **Itinéraires sortants** : un scénario PBX peut également inclure une configuration d'itinéraire sortant qui détermine les numéros que les utilisateurs peuvent composer pour les appels externes.

La commande **make basic-pbx** est utile pour les personnes qui souhaitent exécuter rapidement Asterisk comme un simple **PBX** à des fins de test et d'expérimentation.

Une fois cela fait, vous pouvez personnaliser la configuration générée en fonction de vos besoins et ajouter des fonctionnalités et des extensions plus avancées si nécessaire.

Il convient de rappeler qu' **Asterisk** est une plate-forme très flexible, le scénario PBX n'est donc qu'un exemple parmi de nombreux exemples de configurations pouvant être mises en œuvre.

Pour des applications plus avancées et spécifiques, une configuration plus avancée peut être nécessaire.

```
1: make basic-pbx
```

La dernière commande d'installation créera des fichiers de démarrage.

```
1: make config
```

## 5) Configurer système ASTERISK

On configure le fichier **ASTERISK.CONF**

```
GNU nano 7.2                                asterisk.conf
[options]
; If we want to start Asterisk with a default verbosity for the verbose
; or debug logger channel types, then we use these settings (by default
; they are disabled).
;verbose = 5
;debug = 2

; User and group to run asterisk as. NOTE: This will require changes to
; directory and device permissions.
;runuser = asterisk                ; The user to run as. The default is root.
;rungroup = asterisk              ; The group to run as. The default is root

languageprefix = yes
defaultlanguage = fr
```

On configure le fichier **EXTENSION.CONF**

```
GNU nano 7.2 extensions.conf
!!
!! Automatically generated configuration file
!! Filename: extensions.conf (/etc/asterisk/extensions.conf)
!! Generator: Manager
!! Creation Date: Tue Jun 20 08:31:38 2023
!!
[from-internal]
exten => 6001,1,Dial(PJSIP/alice,10)
exten => 6002,1,Dial(PJSIP/bob,10)
exten => 6003,1,Dial(PJSIP/martin,10) ; Après 10 secondes envoi vers la règle 2
exten => 6099,1,VoiceMailMain() ; 6099 Numéro de téléphone du répondeur

; Règles 2
exten => 6001,2,VoiceMail(6001) ; Appel répondeur compte 6001
exten => 6002,2,VoiceMail(6002) ; Appel répondeur compte 6002
exten => 6003,2,VoiceMail(6003) ; Appel répondeur compte 6003
[general]
static = yes
writeprotect = no
clearglobalvars = yes
[globals]
FEATURES =
DIALOPTIONS =
RINGTIME = 20
FOLLOWMEOPTIONS =
PAGING_HEADER = Intercom
[default]
[macro-stdexten]
exten = s,1,Set(__DYNAMIC_FEATURES=${FEATURES})
exten = s,2,Set(ORIG_ARG1=${ARG1})
exten = s,3,GotoIf("${FOLLOWME_${ARG1}}" = "1")?6:4)
exten = s,4,Dial(${ARG2},${RINGTIME},${DIALOPTIONS})
exten = s,5,Goto(s-${DIALSTATUS},1)

[ Lecture de 136 lignes ]
^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement M-U Annuler
^X Quitter ^R Lire fich. ^_ Remplacer ^U Coller ^J Justifier ^_ Aller ligne M-E Refaire
```

On configure le fichier **PJSIP.CONF** on y ajoutes les utilisateurs

```
GNU nano 7.2 pjsip.conf
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

;Templates for the necessary config sections

[endpoint_internal]()
type=endpoint
context=from-internal
disallow=all
allow=ulaw
language=fr

[auth_userpass]()
type=auth
auth_type=userpass

[aor_dynamic]()
type=aor
max_contacts=1

;Definitions for our phones, using the templates above

[alice](endpoint_internal)
auth=alice
aors=alice
[alice](auth_userpass)
password=alice; put a strong, unique password here instead
username=alice
[alice](aor_dynamic)

[bob](endpoint_internal)

[ Lecture de 47 lignes ]
^G Aide ^O Écrire ^W Chercher ^K Couper ^T Exécuter ^C Emplacement M-U Annuler
^X Quitter ^R Lire fich. ^_ Remplacer ^U Coller ^J Justifier ^_ Aller ligne M-E Refaire
```

On configure le fichier **VOICEMAIL.CONF**

```
GNU nano 7.2 voicemail.conf
; numero messagerie 6099
[general]
format=wav49|gsm|wav|ulaw
[default]
; Numéro de messagerie 6099 => identifiant + mot de passe d'accès à la messagerie, nom d'utilisateur
6001 => 1234, alice
6002 => 1234, bob
6003 => 1234, martin
```

Nous sommes maintenant prêts à exécuter le système Asterisk!!!



## 6) Démarrer le système ASTERISK

Tout d'abord, nous ajouterons le démarrage d'Asterisk au démarrage automatique .

```
1: systemctl enable asterisk.service
```

```
root@debian-markowy2:/usr/src/asterisk-20.4.0# systemctl enable asterisk.service
asterisk.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable asterisk
```

Le message ci-dessus : **asterisk.service** n'est pas un service natif ne devrait pas nous inquiéter, car **Debian** se charge d'ajouter le service au démarrage lui-même.

L'ensemble de commandes suivant nous permettra d'exécuter **Asterisk** et de vérifier son état.

```
1: systemctl start asterisk.service
```

(nota : on peut lancer le système que en tant que sudo ou root)

```
2: systemctl status asterisk.service
```

(nota : on peut vérifier le statut en tant qu'utilisateur simple)

Le démarrage correct du service nous accueillera avec un message similaire au suivant :

```
root@debian-markowy2:/usr/src/asterisk-20.4.0# systemctl status asterisk.service
• asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/etc/init.d/asterisk; generated)
   Active: active (running) since Wed 2023-09-27 14:23:32 CEST; 15s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 31550 ExecStart=/etc/init.d/asterisk start (code=exited, status=0/SUCCESS)
    Tasks: 46 (limit: 2307)
   Memory: 39.3M
      CPU: 538ms
    CGroup: /system.slice/asterisk.service
            └─31562 /usr/sbin/asterisk

Sep 27 14:23:32 debian-markowy2 systemd[1]: Starting asterisk.service - LSB: Asterisk PBX...
Sep 27 14:23:32 debian-markowy2 asterisk[31550]: Starting Asterisk PBX: asterisk.
Sep 27 14:23:32 debian-markowy2 systemd[1]: Started asterisk.service - LSB: Asterisk PBX.
```

Nous pouvons maintenant lancer la console Asterisk et commencer à surveiller le service et sa configuration ultérieure. Amusez-vous à travailler avec un astérisque.

```
1: sudo asterisk -rvvvvv
```

```
laurent@debian:/etc/asterisk$ asterisk -rvvvvvvv
-bash: asterisk: commande introuvable
laurent@debian:/etc/asterisk$ sudo asterisk -rvvvvv
Asterisk 20.7.0, Copyright (C) 1999 - 2022, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 20.7.0 currently running on debian (pid = 1741)
debian*CLI> _
```

L'installation d'**ASTERISK** est terminée maintenant faut configurer **ASTERISK** et les Postes des Utilisateurs.

## 6) Installation des deux logiciels de communication

On a choisis **LINPHONE** et **MICROSIP**

### 1.0 installation

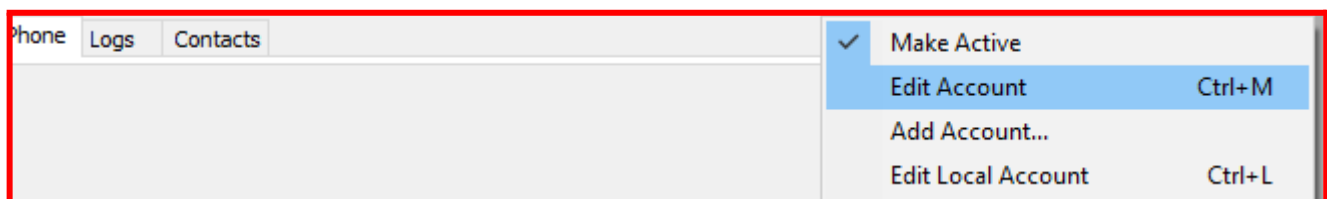
On télécharge sur les sites : les deux logiciels

<https://www.linphone.org/>

<https://www.microsip.org/>

Dézipper le fichier portable.zip dans le répertoire de votre choix, puis lancez *microsip.exe*.

On va prendre MICROSIP comme exemple pour comprendre l'installation et la configuration du logiciel de mise en relation client.



### 2.0 configuration

Définissez le Serveur SIP (serveur nativIP), le Proxy notamment si vous utilisez le port 5080 pour les postes internes (config. par défaut de nativIP serveur).

Sur cet exemple, le poste 6002 est enregistré sur le serveur 192.168.126.140. Le nom présenté est BOB.

Nom du compte	bob	
Serveur SIP	192.168.126.140	?
Proxy SIP		?
Nom d'utilisateur *	bob	?
Domaine *	192.168.126.140	?
Login	bob	?
Mot de passe	***	?
Nom à afficher		?
N° de la boîte vocale		?
Préfixe d'appel		?
Plan de numérotation		?
<input type="checkbox"/> Hide Caller ID		?
Chiffrement	Désactivé	?
Transport	UDP	?
Adresse publique	Auto	?
Actualiser l'enregist...	300	Signalisation 15
<input type="checkbox"/> Afficher ma présence		?
<input type="checkbox"/> Autoriser la réécriture de l'IP		?
<input type="checkbox"/> ICE		?
<input type="checkbox"/> Désactiver les minuteurs de session		?

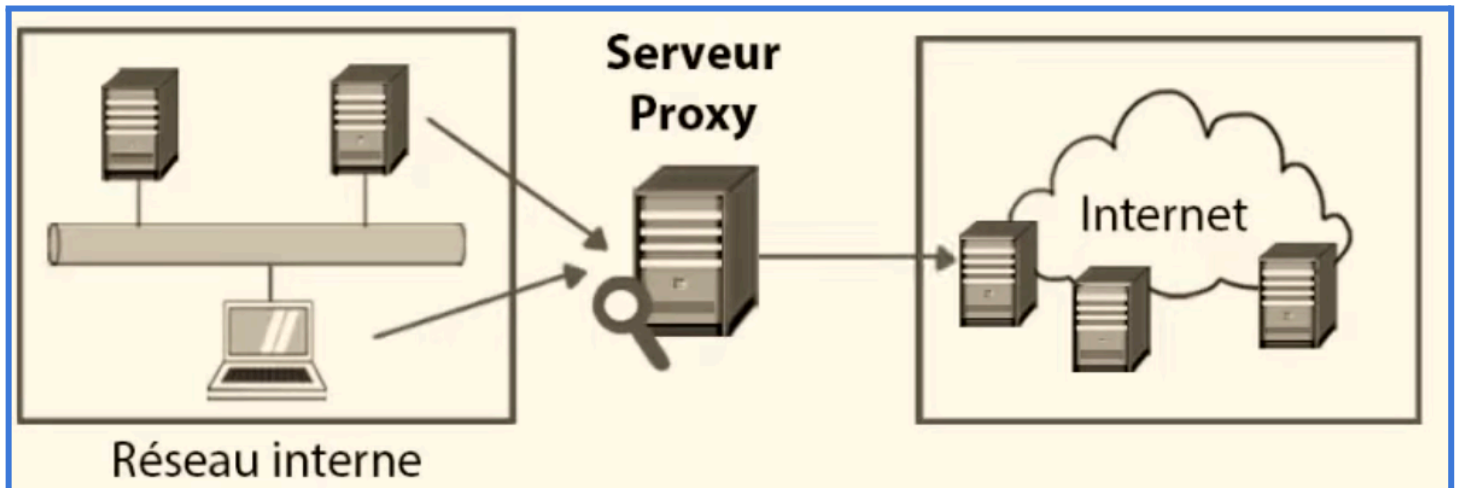
[X] [Sauvegarder] [Annuler]

Puis on l'install comme tous logiciels en cliquant sur suivant et terminer

## ATTENTION : Le transport doit être forcé sur UDP.

### POUR INFO:

Le serveur proxy est un serveur intermédiaire qui va permettre à un utilisateur d'accéder à internet



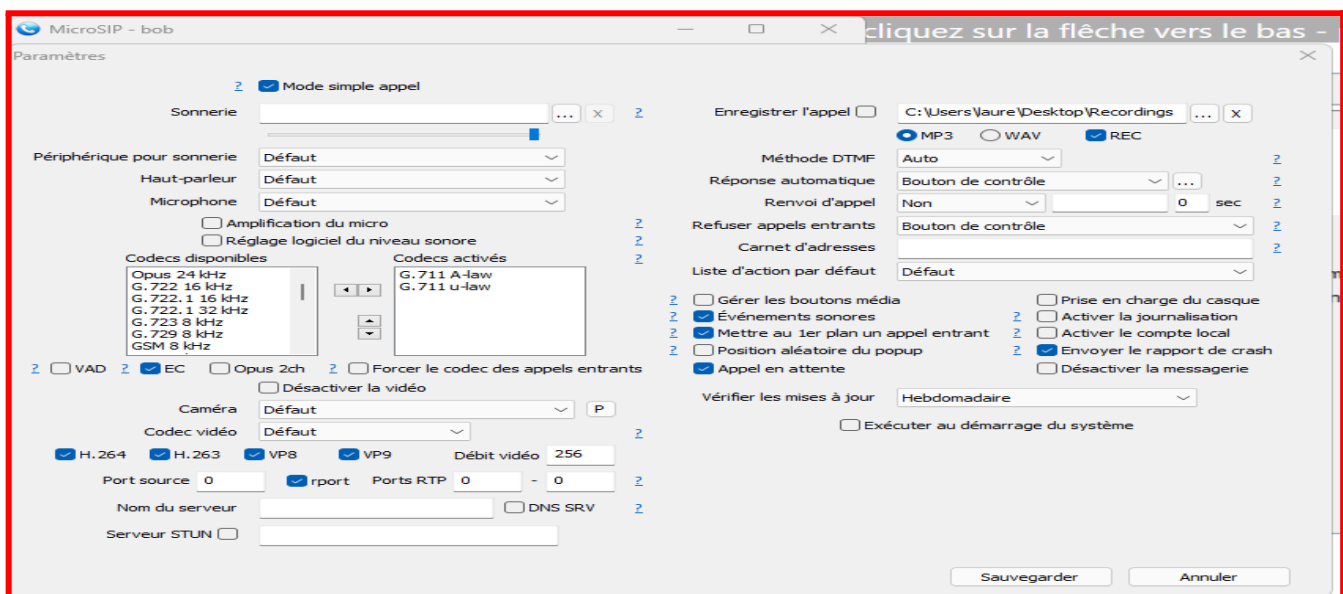
L'utilisateur envoie sa requête au serveur proxy qui lui-même va faire la requête au serveur distant.

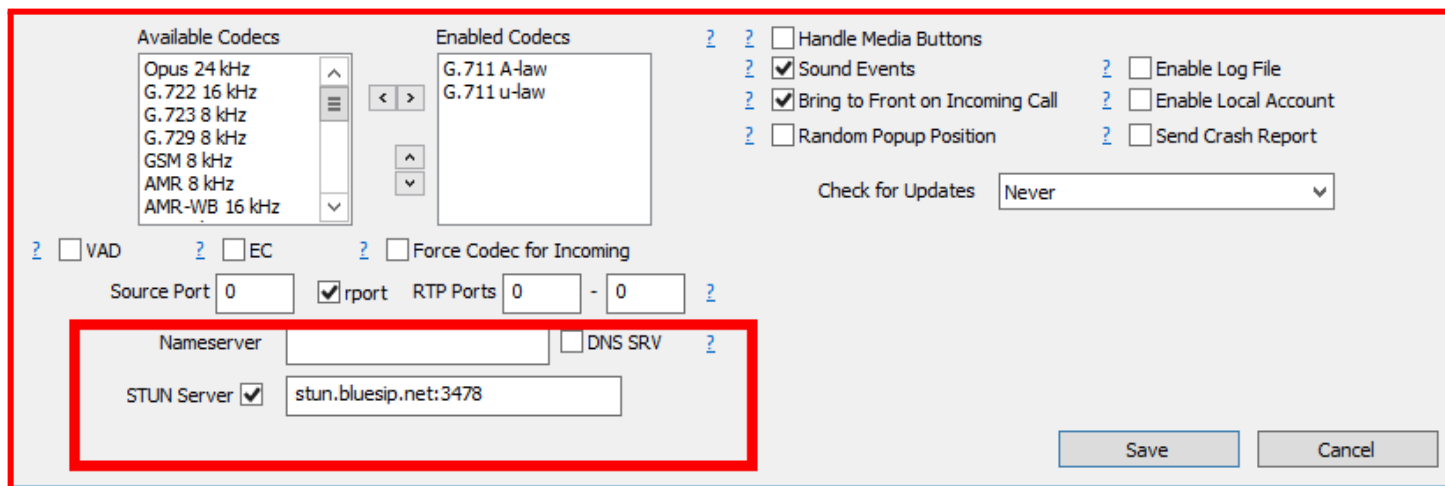
- 1° Filtrer les accès internet au sein d'une entreprise
- 2° Être Anonyme sur un réseau en masquant l'adresse de l'utilisateur mais qui sera connu du proxy
- 3° Accélérer la navigation via des fonctions : compression de données, le filtrage des contenus lourds et aussi la fonction de cache. (mémoriser les données pour éviter de re-télécharger des données et donc accélérer l'affichage et aussi de mémoriser les pages les plus visitées pour les proposer aux internautes le plus rapidement possible).
- 4° Il enregistre les requêtes et donc crée des fichiers log pour la traçabilité du réseau.
- 5° Filtrer le réseau avec des **Whites List** (filtrage de site autorisée) ou **Black List** (filtrage de site non autorisée).

## 3 - (optionnel)

Selon votre environnement réseau, il peut être nécessaire d'activer STUN pour utiliser votre IP publique (Si vous êtes en télétravail sans VPN par exemple).

Dans la configuration (cliquez sur la flèche vers le bas - puis Settings)





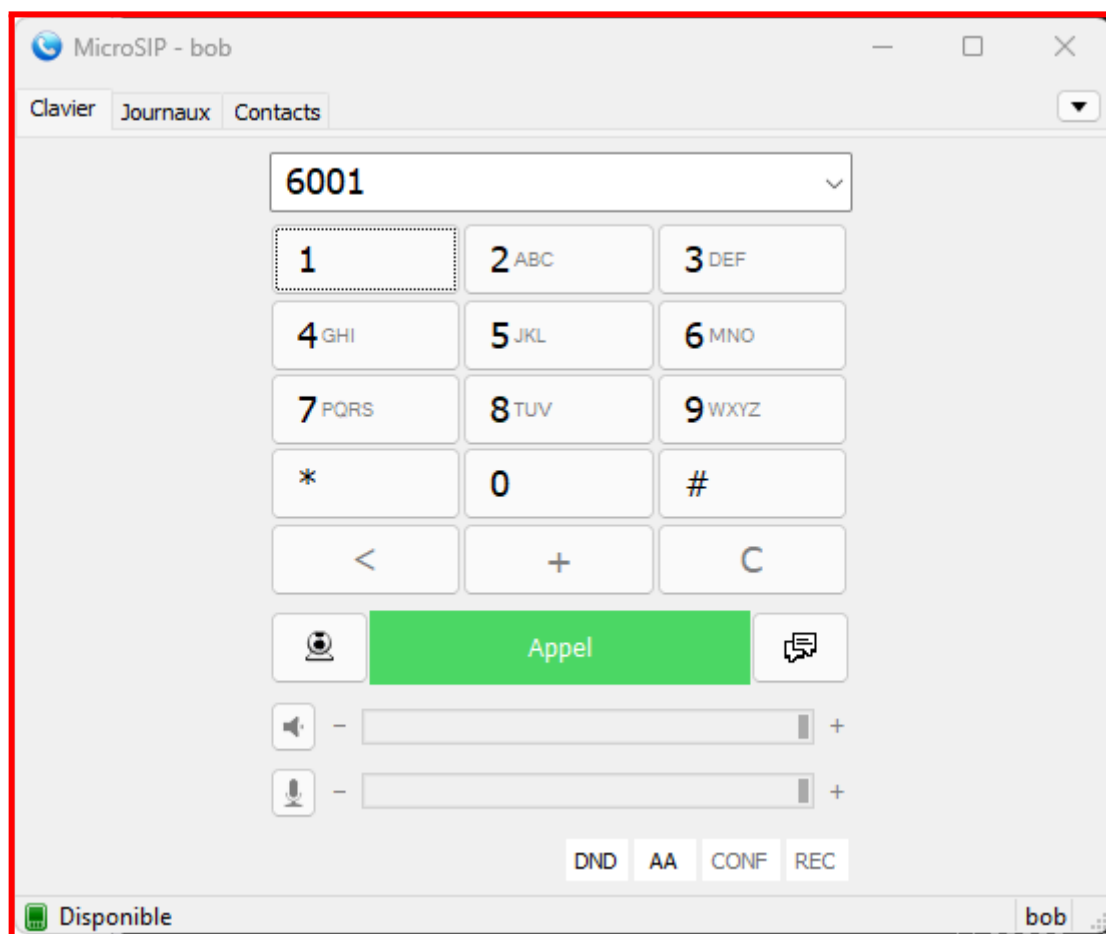
[Mon IP publique : Définitions](#)

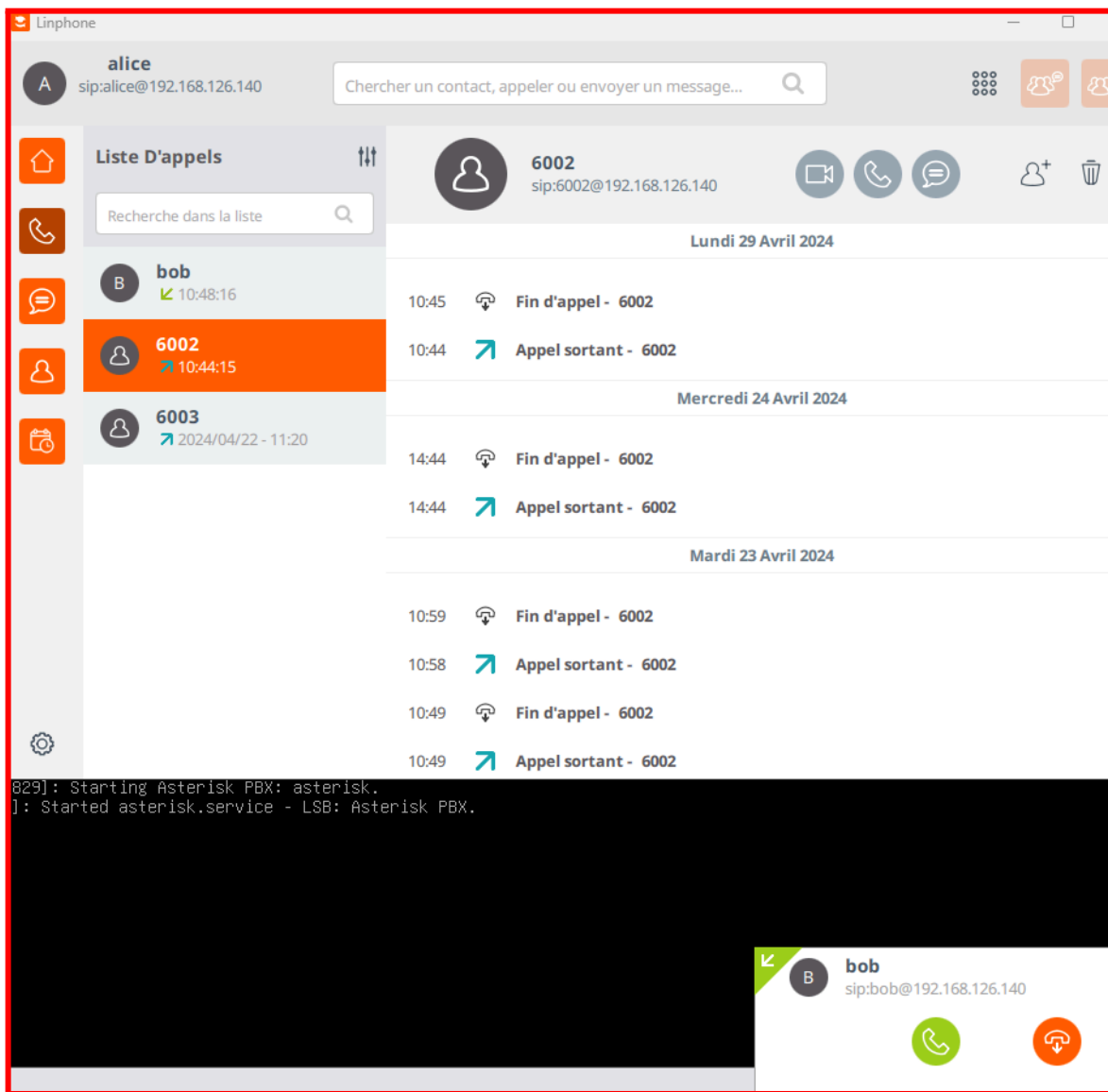
### 3.0 utilisation

Une fois l'installation terminée, un indicateur vert Online (en bas à gauche) vous confirme que vous êtes enregistré.

Vous pouvez appeler le numéro de votre choix - ici 6001 (poste, service vocal....) en cliquant sur Call (ou sur la webcam pour un appel vidéo).

Dans la configuration (cliquez sur la flèche vers le bas - puis Settings)





Dans l'image ci dessus **BOB** appel **Alice** on voit bien en bas à droite l'appel de **BOB** donc les deux fonctionnent correctement.

#### 4.0 Renseignez-vous sur les concepts et les protocoles utilisés par les serveurs VoIP:

La VoIP (Voice over Internet Protocol) est une technologie qui permet de transmettre des communications vocales via Internet plutôt que par les réseaux de téléphonie traditionnels.

Les serveurs VoIP jouent un rôle crucial dans cette infrastructure, assurant le routage des appels, la gestion des utilisateurs, et la conversion des signaux vocaux en données numériques et vice versa.

Voici quelques concepts et protocoles clés utilisés par les serveurs **VoIP** :

##### 1. Protocoles de signalisation :

- **SIP (Session Initiation Protocol)** : SIP est le protocole de signalisation le plus couramment utilisé pour établir, modifier et terminer les sessions de communication VoIP. Il gère la signalisation des appels, la localisation des utilisateurs et la négociation des paramètres de session.
- **H.323** : Bien que moins utilisé que SIP, H.323 est un protocole de signalisation plus ancien qui est également utilisé pour la VoIP. Il fournit des fonctionnalités similaires à SIP mais utilise un ensemble différent de spécifications techniques.

## 2. Protocoles de transport :

- **RTP (Real-Time Transport Protocol)** : RTP est utilisé pour le transport des données audio et vidéo en temps réel sur les réseaux IP. Il travaille en tandem avec le protocole RTCP (Real-Time Control Protocol) pour surveiller la qualité de service et fournir des informations de contrôle de la transmission.
- **TCP (Transmission Control Protocol)** et **UDP (User Datagram Protocol)** : Ces protocoles sont utilisés pour transporter les données de signalisation et de médias entre les clients VoIP et les serveurs. TCP est souvent utilisé pour la signalisation tandis que UDP est préféré pour le transport des flux audio et vidéo en raison de sa faible latence et de son efficacité.

## 3. Gestion des sessions :

- **SDP (Session Description Protocol)** : SDP est utilisé pour décrire les paramètres des sessions multimédias, y compris les codecs audio et vidéo utilisés, les adresses IP et les ports de communication. Il est souvent échangé entre les clients et les serveurs VoIP lors de la phase de négociation de session.
- **ICE (Interactive Connectivity Establishment)** : ICE est un ensemble de techniques utilisées pour établir des connexions peer-to-peer entre les clients VoIP, souvent utilisées pour contourner les pare-feu et les routeurs NAT afin de permettre des communications directes.

## 4. Gestion des utilisateurs :

- **LDAP (Lightweight Directory Access Protocol)** : LDAP est un protocole utilisé pour interroger et modifier les annuaires d'utilisateurs, qui sont souvent utilisés pour stocker des informations d'authentification et de configuration pour les utilisateurs VoIP.
- **RADIUS (Remote Authentication Dial-In User Service)** : RADIUS est un protocole d'authentification et d'autorisation largement utilisé dans les réseaux VoIP pour contrôler l'accès des utilisateurs et appliquer des politiques de sécurité.

En résumé, les serveurs VoIP reposent sur un ensemble complexe de protocoles et de technologies pour permettre des communications vocales efficaces et fiables sur les réseaux IP.

Ces protocoles jouent des rôles spécifiques dans la signalisation, le transport des médias, la gestion des sessions et des utilisateurs, assurant ainsi le bon fonctionnement des services VoIP

## 7.0 Trouvez un protocole pour sécuriser/chiffrer les appels transitant par le serveur (SBC Session border controllers?) :

Un protocole couramment utilisé pour sécuriser et chiffrer les appels transitant par un serveur VoIP, y compris un Session Border Controller (SBC), est le protocole **SRTP** (Secure Real-time Transport Protocol).

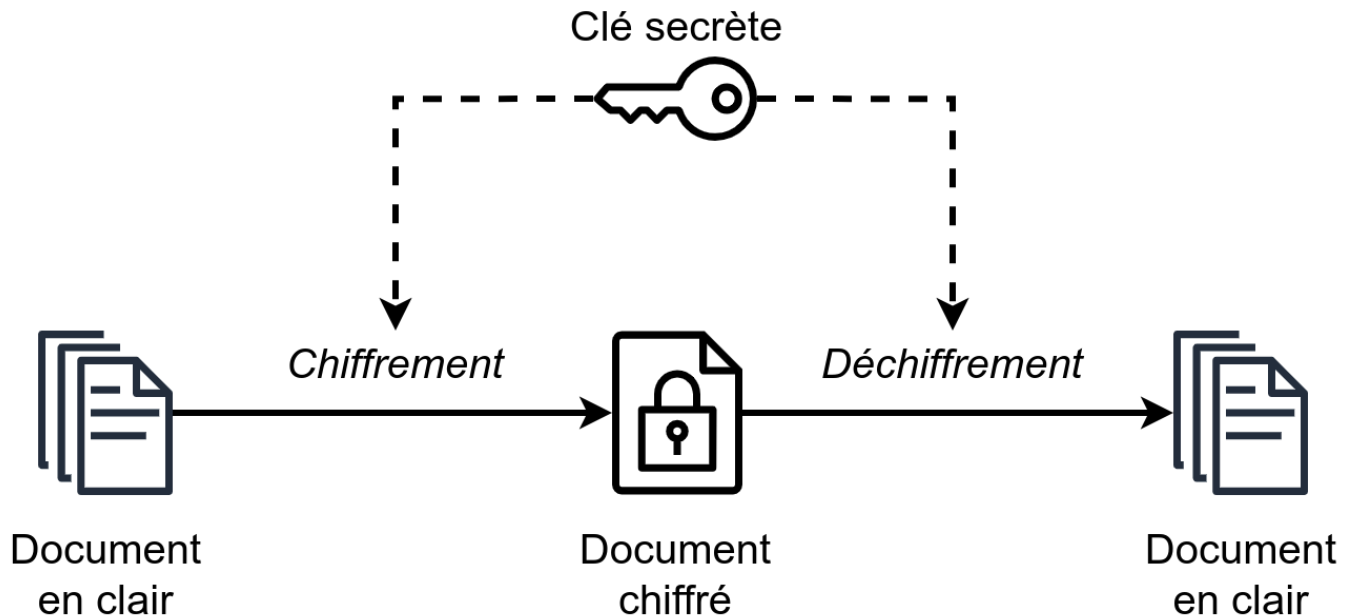
**SRTP** est une extension du protocole **RTP** (Real-time Transport Protocol) qui fournit des mécanismes de confidentialité, d'intégrité et d'authentification pour les flux de médias en temps réel.

Voici quelques caractéristiques clés de **SRTP** :

1. **Chiffrement des données** : **SRTP** chiffre les données audio et vidéo en utilisant des algorithmes de chiffrement symétrique, tels que **AES** (Advanced Encryption Standard), pour garantir que les conversations restent confidentielles et sécurisées contre l'interception.
2. **Intégrité des données** : **SRTP** utilise des codes d'authentification des messages (**MAC**) pour détecter toute altération des données pendant le transport, assurant ainsi l'intégrité des flux de médias.
3. **Protection contre les attaques par replay** : **SRTP** intègre des mécanismes pour prévenir les attaques par replay, où un attaquant tente de réutiliser des paquets de données précédemment capturés pour compromettre la sécurité de la communication.

4. **Authentification des participants** : **SRTP** permet l'authentification des participants à l'aide de mécanismes de clés partagées ou de certificats numériques, garantissant que seuls les utilisateurs autorisés peuvent accéder aux flux de médias sécurisés.

En utilisant **SRTP**, les serveurs **VoIP**, y compris les **SBC**, peuvent garantir la confidentialité et l'intégrité des communications vocales, ce qui est essentiel dans les environnements où la sécurité des données est une préoccupation majeure, tels que les entreprises et les fournisseurs de services **VoIP**



Pour activer le **SRTP** sous **ASTERISK** faut l'installer (en cas d'oubli au départ de la configuration) :

```
.1 : sudo aptinstall libsrtp2-dev
```

puis

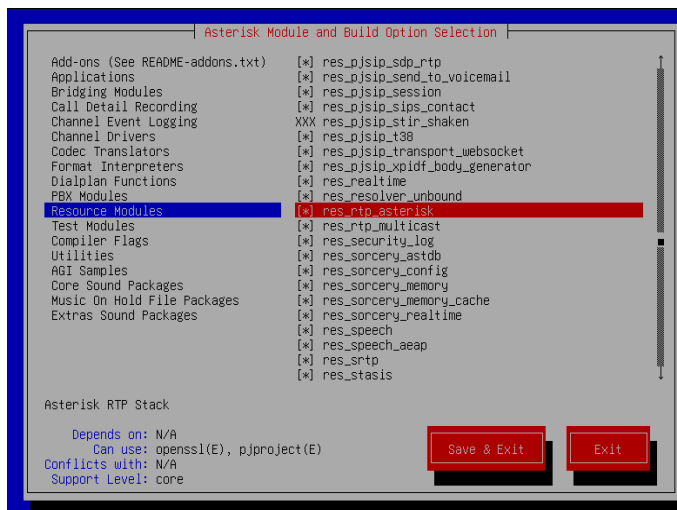
dirigez vous vers le dossier ou est stocké le MENU SELECT:

```
.2 : /usr/src/asterisk-20.7.0
```

puis

ouvrir la fenêtre de configuration avec

```
.3 : sudo make menuselect
```



Dans ce menu faut cocher tout ce qui concerne module **SRTP**

Sauvegarder Quitter

puis  
faire un

```
sudo make
```

(pour que soit pris en compte les modifications du **MENU SELECT**).

puis réinstaller Asterisk faire :

```
sudo make install
```

puis faut activer les modules demandé

faut allez vers le répertoire ou est installer les fichiers de configuration d'Asterisk

```
sudo nano /etc/asterisk/modules.conf
```

rajouter cette ligne pour **activer** le module **SRTP** => `load = res_srtp.so`

```
GNU nano 7.2 /etc/asterisk/modules.conf *
load = res_pjsip_outbound_authenticator_digest.so
load = res_pjsip_outbound_publish.so
load = res_pjsip_outbound_registration.so
load = res_pjsip_path.so
load = res_pjsip_pidf_body_generator.so
load = res_pjsip_pidf_digium_body_supplement.so
load = res_pjsip_pidf_eyeball_body_supplement.so
load = res_pjsip_publish_asterisk.so
load = res_pjsip_pubsub.so
load = res_pjsip_refer.so
load = res_pjsip_registrar.so
load = res_pjsip_rfc3326.so
load = res_pjsip_sdp_rtp.so
load = res_pjsip_send_to_voicemail.so
load = res_pjsip_session.so
load = res_pjsip.so
load = res_pjsip_t38.so
load = res_pjsip_transport_websocket.so
load = res_pjsip_xpidf_body_generator.so
load = res_rtp_asterisk.so
load = res_sorcery_astdb.so
load = res_sorcery_config.so
load = res_sorcery_memory.so
load = res_sorcery_realtime.so
load = res_timing_timerfd.so
load = res_srtp.so
; Do not load res_hep and kin unless you are using HEP monitoring
; <http://sipcapture.org> in your network.

noload = res_hep.so
noload = res_hep_pjsip.so
noload = res_hep_rtp.so

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C EmplacementM-U Annuler
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligneM-E Refaire
```



Pour vérifier les modules actifs faut taper toujours depuis le CLI :

`module show`

```
debian*CLI> module show_
```

Sa affiche les modules actifs pour **asterisk**

```
core
res_pjsip_registrar.so      PJSIP Registrar Support      0      Running
core
res_pjsip_rfc3326.so        PJSIP RFC3326 Support        0      Running
core
res_pjsip_sdp_rtp.so        PJSIP SDP RTP/AVP stream handler 0      Running
core
res_pjsip_send_to_voicemail.so PJSIP REFER Send to Voicemail Support 0      Running
core
res_pjsip_session.so        PJSIP Session resource      15     Running
core
res_pjsip_t38.so           PJSIP T.38 UDPTL Support     0      Running
core
res_pjsip_transport_websocket.so PJSIP WebSocket Transport Support 0      Running
core
res_pjsip_xpidsf_body_generator.so PJSIP Extension State PIDF Provider 0      Running
core
res_rtp_asterisk.so         Asterisk RTP Stack           0      Running
core
res_sorcery_astdb.so        Sorcery Astdb Object Wizard   3      Running
core
res_sorcery_config.so       Sorcery Configuration File Object Wizard 16     Running
core
res_sorcery_memory.so       Sorcery In-Memory Object Wizard 2      Running
core
res_sorcery_realtime.so     Sorcery Realtime Object Wizard 0      Running
core
res_srtp.so                 Secure RTP (SRTP)            0      Running
core
res_timing_timerfd.so       Timerfd Timing Interface     0      Running
core
sounds                       Sounds Index                  1      Running
core
udptl                       UDPTL                        2      Running
core
96 modules loaded
```

puis `verifions`

## 8.0 État initial du système:

Vérifions que le protocole **SRTP** est bien actif pour protéger les communications, on le lit dans l'interface **CLI** au moment des appels:

faut d'abord dans l'interface CLI augmenter le verbose pour en lire le détails des protocoles de communications avec la commande => `-rvvvvv`

donc la commande est => `asterisk -rvvvvv`

## DONC :

Lors d'un appel il doit y avoir le détail des appels et voir si le protocole **SRTP** est bien actif et opérationnel :

```
laurent@debian:~$ sudo asterisk -rvvvv
Asterisk 20.7.0, Copyright (C) 1999 - 2022, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 20.7.0 currently running on debian (pid = 915)
-- Executing [6001@from-internal:1] Dial("PJSIP/bob-00000016", "PJSIP/alice,10") in new stack
-- Called PJSIP/alice
-- PJSIP/alice-00000017 is ringing
> 0x7f96880d9f80 -- Strict RTP learning after remote address set to: 192.168.126.1:56727
-- PJSIP/alice-00000017 answered PJSIP/bob-00000016
> 0x7f96880b7700 -- Strict RTP learning after remote address set to: 10.10.10.132:4026
-- Channel PJSIP/alice-00000017 joined 'simple_bridge' basic-bridge <db3aac16-873a-43c7-a8fd-23a22b52588d>
-- Channel PJSIP/bob-00000016 joined 'simple_bridge' basic-bridge <db3aac16-873a-43c7-a8fd-23a22b52588d>
> Bridge db3aac16-873a-43c7-a8fd-23a22b52588d: switching from simple_bridge technology to native_rtp
> Remotely bridged 'PJSIP/bob-00000016' and 'PJSIP/alice-00000017' - media will flow directly between them
> 0x7f96880b7700 -- Strict RTP learning after remote address set to: 10.10.10.132:4026
> 0x7f96880d9f80 -- Strict RTP switching to RTP target address 192.168.126.1:56727 as source
> 0x7f96880b7700 -- Strict RTP switching to RTP target address 10.10.10.132:4026 as source
-- Channel PJSIP/alice-00000017 left 'native_rtp' basic-bridge <db3aac16-873a-43c7-a8fd-23a22b52588d>
-- Channel PJSIP/bob-00000016 left 'native_rtp' basic-bridge <db3aac16-873a-43c7-a8fd-23a22b52588d>
== Spawn extension (from-internal, 6001, 1) exited non-zero on 'PJSIP/bob-00000016'
debian*CLI>
```

On voit bien ci dessus la clé et le protocole **Strict RTP learning** et **switching** c'est sécurisé !!

## 9.0 Fonctionnalité testée / périmètre du test :

### 1. Test d'écho :

Utilisez la fonctionnalité d'écho d'Asterisk pour tester la qualité de la voix. Vous pouvez placer un appel depuis un téléphone vers un numéro interne configuré pour répondre automatiquement et renvoyer l'écho de la voix. Cela vous permet de vérifier la latence, l'écho et la clarté de la voix.

### 2. Test de performance du réseau :

Utilisez des outils de test de performance réseau tels que `iperf` pour mesurer la bande passante, la latence et la gigue (variation de latence) entre les nœuds de votre réseau, y compris votre serveur Asterisk et les périphériques utilisateurs.

### 3. Test de MOS (Mean Opinion Score) :

Utilisez des outils logiciels ou en ligne pour effectuer un test de MOS. Le MOS est une mesure subjective de la qualité de la voix perçue par les utilisateurs finaux. Des outils tels que `PESQ`

(Perceptual Evaluation of Speech Quality) peuvent simuler des appels et fournir un score MOS basé sur la qualité perçue.

#### 4. Surveillance de la performance du serveur :

Surveillez les métriques de performance de votre serveur Asterisk à l'aide d'outils de surveillance tels que `top`, `htop` ou des solutions de surveillance système plus avancées. Assurez-vous que le serveur fonctionne efficacement et qu'il n'y a pas de goulots d'étranglement dans les ressources telles que le processeur, la mémoire ou l'utilisation du disque.

#### 5. Analyse des logs :

Analysez les logs d'Asterisk pour détecter les erreurs, les avertissements ou les anomalies liées à la qualité de la voix. Vous pouvez rechercher des indications telles que des erreurs de codec, des déconnexions ou des appels abandonnés.

#### 6. Utilisation d'outils tiers :

Explorez des solutions tierces conçues spécifiquement pour le test et la surveillance de la qualité de la voix dans les systèmes de téléphonie IP. Ces solutions peuvent offrir des fonctionnalités avancées telles que des rapports détaillés, des graphiques de performances et des recommandations d'amélioration.

En utilisant une combinaison de ces méthodes, vous pouvez évaluer différents aspects de la qualité de la ligne de votre installation téléphonique Asterisk et identifier les éventuels problèmes ou domaines d'amélioration.

## 10.0 (Résultat du test : OK/NOK)

### Réussite du test

```
bash Copy code  
  
Résultat du test : OK
```

### Echec du test

```
bash Copy code  
  
Résultat du test : NOK
```

# 11.0 Automatisation

Mise en place d'un menu automatique (IVR) sur Asterisk :

## 1. Définition des menus :

- **Menu principal :**
  - Appuyez sur 1 pour le service comptabilité.
  - Appuyez sur 2 pour le service des ressources humaines.
  - Appuyez sur 3 pour parler à un représentant du service client.
- **Service comptabilité :**
  - Appuyez sur 1 pour les factures impayées.
  - Appuyez sur 2 pour les demandes de remboursement.
- **Service des ressources humaines :**
  - Appuyez sur 1 pour les questions relatives aux RH.
  - Appuyez sur 2 pour les offres d'emploi.
- **Service client :**
  - Appuyez sur 1 pour le support technique.
  - Appuyez sur 2 pour le service après-vente.

```
[menu_principal]
exten => 1234,1,Answer()
    same => n,Background(menu_principal)
    same => n,WaitExten(10)

exten => 1,1,Goto(service_comptabilite,s,1)
exten => 2,1,Goto(service_rh,s,1)
exten => 3,1,Goto(service_client,s,1)
;[menu_principal]
;exten => 1,1,Playback(service-comptabilite)
;same => n,Goto(service_comptabilite,s,1)
;exten => 2,1,Playback(service-rh)
;same => n,Goto(service_rh,s,1)
;exten => 3,1,Playback(service-client)
;same => n,Goto(service_client,s,1)

[service_comptabilite]
exten => s,1,BackgroundMenu(factures-impayees)
exten => 1,1,Playback(factures-impayees)
; Logique de traitement des appels pour les factures impayées
exten => 2,1,Playback(demandes-remboursement)
; Logique de traitement des appels pour les demandes de remboursement

[service_rh]
exten => s,1,BackgroundMenu(questions-rh)
exten => 1,1,Playback(questions-rh)
```

## CREATION DU MESSAGE VOCAL EN FRANCAIS avec GTTS et SOX

Installer les paquets python 3 pour préparer GTTS qui va servir a transformé un message écrit en vocal

`sudo apt install python3-pip`

Vérifier la version installer de python 3

`pip3 --version`

**A**vec GGTS qui permet de convertir un texte en vocal faut installer les paquets:

`sudo pip install gTTS` cette ligne ne fonctionne pas faut le forcer avec  
=> `sudo pip install gTTS --break-system-packages`

**U**ne fois GTTS installé faut créer la ligne de commande vocal =>

`sudo gtts-cli 'bienvenue sur le repondeur de darty, pour le menu 1 tapez 1, pour le menu 2 tapez 2, pour le menu 3 tapez 3 sinon barrez vous' -l fr --output menuauto.mp3`

```
gtts-cli 'bienvenue sur le repondeur de darty, pour le menu 1 tapez 1, pour le menu 2 tapez 2, pour le menu 3 tapez 3 sinon barrez vous' -l fr --output menuauto.mp3
sudo gtts-cli 'bienvenue sur le repondeur de darty, pour le menu 1 tapez 1, pour le menu 2 tapez 2, pour le menu 3 tapez 3 sinon barrez vous' -l fr --output menuauto.mp3
ls
```

**C**onvertir le fichier **.mp3** en **.Ulaw** avec **SOX** qui est un convertisseur de son

Installer les paquets **SOX** :

`sudo apt-get install perl libwww-perl sox mpg123`

Installer librairie **MP3** de **SOX**:

`sudo apt install sox libsox-fmt-mp3`

**C**onvertir le fichier **.mp3** en **.ulaw**

`sudo sox -V menuauto.mp3 -r 8000 -c 1 -t ul menuauto.ulaw`

faut que le fichier audio soit dans le dossier

`/var/lib/asterisk/sounds/fr`