

# Implémentation d'une Solution de Téléphonie IP avec Asterisk

## L'objectif du projet :

Déployer un serveur de téléphonie sur IP sur une machine virtuelle Debian et configurer un client VoIP sur des smartphones, permettant ainsi d'établir des communications vocales à travers le serveur.

## Déroulement du projet :

Pour la réalisation de notre projet, nous avons suivi les étapes suivantes :

1. **Installation** d'une machine virtuelle Debian sans interface graphique.
2. **Installation d'Asterisk** sur Debian.
3. **Configuration d'Asterisk.**
4. **Création de deux utilisateurs** : [1001] et [1002].
5. **Configuration des utilisateurs** dans les fichiers d'Asterisk:
  - nano /etc/asterisk/extensions.conf
  - nano /etc/asterisk/pjsip.confCes configurations ont permis aux utilisateurs de pouvoir s'appeler entre eux.
6. **Mise en place d'une messagerie interactive.**
7. **Téléchargement de l'application PortSIP** sur les smartphones.
8. **Phase de test** : réalisation d'appels entre deux smartphones via l'application PortSIP, avec un **résultat positif**

## **VoIP**

### **Qu'est ce que la Téléphonie sur IP VoIP?**

VoIP est un acronyme qui signifie Voice Over Internet Protocole, ou en d'autres termes, la transmission de la voix via Internet. C'est une technologie qui permet de délivrer des communications vocales ou multimédia (vidéo par exemple) via le réseau Internet (IP).

### **Comment fonctionne la téléphonie IP ?**

La téléphonie sur IP (ou VoIP pour Voix sur IP) est un mode de téléphonie utilisant le protocole de télécommunications créé pour Internet (IP). La voix est numérisée puis acheminée sous forme de paquets comme n'importe quelles autres données.

### **Quelques avantages sur la téléphonie sur IP**

Le faible coût: L'utilisation du réseau internet permet à la VoIP de fonctionner avec des coûts très faibles. Ainsi, même s'il est important de s'assurer de la qualité du réseau, la téléphonie IP permet à ses utilisateurs de faire des économies substantielles par rapport au système traditionnel.

Une grande flexibilité d'installation :

L'un des intérêts majeur de la téléphonie IP est qu'elle se branche sur le réseau internet ce qui permet aux utilisateurs d'utiliser leur téléphone de n'importe quel endroit du monde sans changer de numéro. La contrainte est d'être connecté au Web.

Une grande facilité d'utilisation :

L'utilisation d'un système numérique permet aux terminaux d'intégrer et de mixer un grand nombre de mode de communication et de supports différents (téléphones, visioconférence, ordinateurs, etc...).

Une offre de services complémentaires:

La numérisation des données permet un traitement informatique et par conséquent une grande amélioration des fonctions proposées : enregistrement des appels, statistiques d'appels, regroupement de plusieurs établissements distants sur un même standard, envoi des messages vocaux par mail, connexion du télécopieur (scanner, mail, ...), gestion des horaires, ...

Gain en Mobilité :

Avec des postes qui ne sont plus physiquement reliés à des lignes, la VoIP permet à l'utilisateur de conserver son numéro dans ses déplacements (par exemple travailler de la maison en affichant le numéro du bureau!).

## ASTERISK

**Asterisk** c'est un logiciel open source qui transforme un simple ordinateur en un standard téléphonique capable de gérer des appels via Internet.

**Asterisk** permet de passer des appels téléphoniques sur un réseau informatique au lieu d'utiliser une ligne téléphonique classique.

Au lieu de passer par des câbles téléphoniques traditionnels, Asterisk transforme la voix en "petits paquets de données" et les envoie via le réseau.

Asterisk a été conçu en 1999 par Mark Spencer, le fondateur de la société américaine Digium. Au départ, il a été développé pour répondre aux besoins de Spencer, qui cherchait une solution téléphonique économique pour son entreprise.

Le nom Asterisk vient du symbole astérisque (\*) utilisé dans la numérotation à fréquences vocales multiples (DTMF), un système qui permet d'envoyer des signaux sonores à travers les lignes téléphoniques. Ce symbole est souvent utilisé sur les téléphones pour des fonctions spéciales ou des raccourcis.

Asterisk est...

- un logiciel **open source**
- écrit en **langage de programmation C**
- fonctionnant sous **Linux** (ou d'autres types de **Unix**)

### Que peut-on faire avec Asterisk ?

1. **Passer et recevoir des appels** : entre différents ordinateurs ou téléphones IP connectés au serveur.
2. **Créer un standard téléphonique** : pour une entreprise avec des numéros internes (ex: 1001 pour Alice, 1002 pour Bob).
3. **Ajouter des options avancées** : comme des messageries vocales, des menus interactifs (« Tapez 1 pour parler à un agent, tapez 2 pour le support... »).
4. **Enregistrer les appels**, mettre en attente avec de la musique, transférer les appels, etc.

**Appels internes** (entre des collègues dans une entreprise).

**Appels vers l'extérieur** (vers des téléphones classiques).

**Conférences** (plusieurs personnes dans le même appel).



## Les avantages de la téléphonie IP sous Asterisk

Parce que le logiciel d'Asterisk est open source, Asterisk peut fondamentalement être adapté à des besoins propres.

Les PBX traditionnels nécessitent souvent de matériel supplémentaire pour les extensions, ce qui peut rapidement augmenter le prix de l'installation. En revanche, Asterisk est déjà une solution logicielle sans la contrepartie des blocs matériels traditionnels.

Il permet également la connexion de lignes téléphoniques analogiques via des passerelles adaptées, de sorte que même les appareils plus anciens peuvent continuer à être utilisés.






## Les inconvénients d'Asterisk

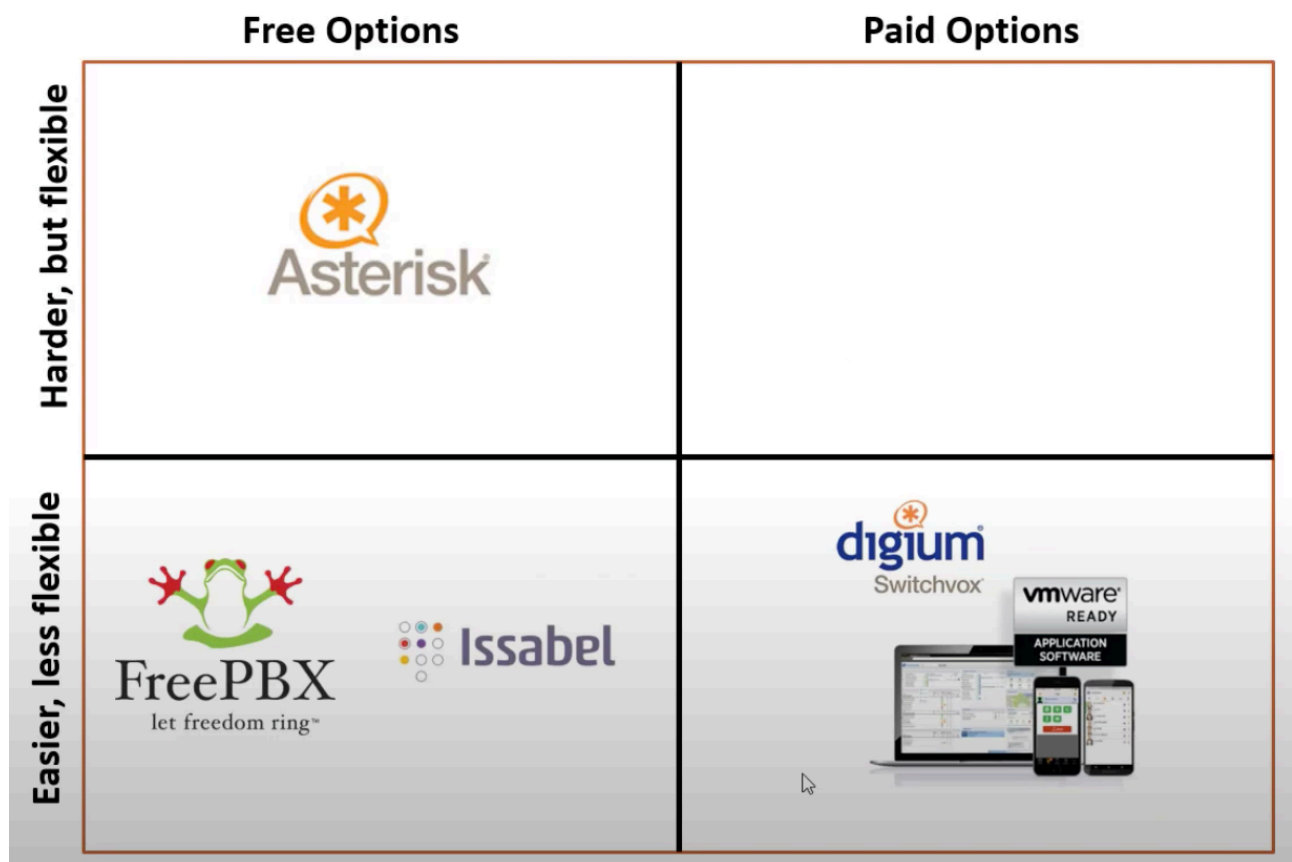
La programmation et la maintenance du système téléphonique lie des ressources importantes à l'entreprise alors qu'on ne "voulait seulement qu'appeler". Asterisk est plus complexe mais flexible, ce qui signifie que vous devez tout configurer manuellement. Il n'y a pas d'interface graphique.

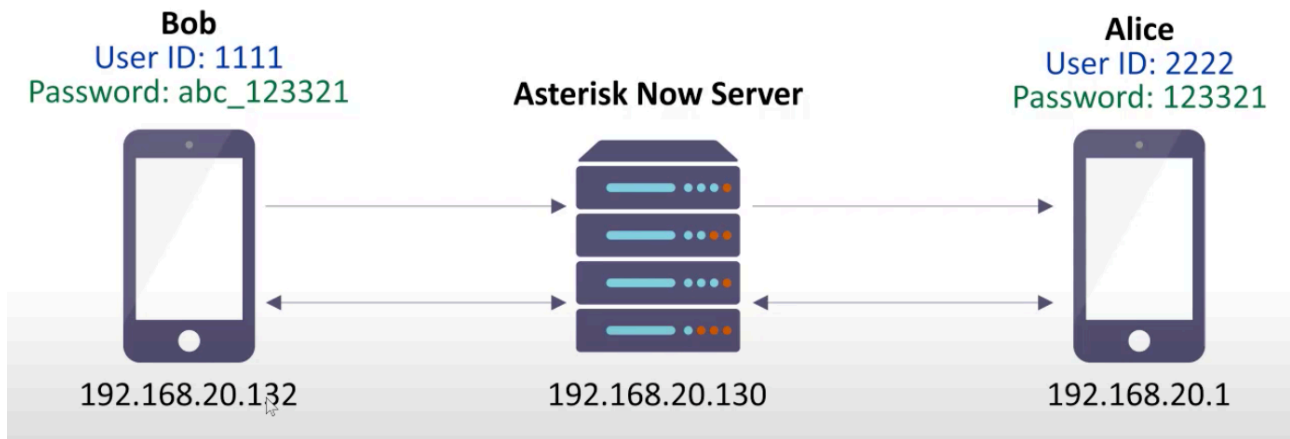
**Il est puissant, gratuit et personnalisable**, mais nécessite des compétences en administration système et réseau.

## Asterisk est-il obligatoire ?

**Non**, Il existe **d'autres solutions** pour gérer la VoIP:

Solution	Type	Exemple	Open-source ou payant ?
<b>Asterisk</b>	PABX IP	Asterisk	Open-source (gratuit) 
<b>FreePBX</b>	Interface graphique basée sur Asterisk	FreePBX	Open-source (mais avec des options payantes) 
<b>3CX</b>	PABX IP clé en main	3CX	Payant 
<b>Cisco CallManager</b>	PABX IP pour grandes entreprises	Cisco CUCM	Payant 
<b>Opérateur Cloud VoIP</b>	Service VoIP géré par un fournisseur	RingCentral, OVH VoIP, Keyyo	Payant 





## Protocoles utilisés par les serveurs VoIP

SIP (Session Initiation Protocol)

SIP c'est un **ensemble de règles (protocole)** qui permet aux téléphones VoIP de communiquer entre eux.

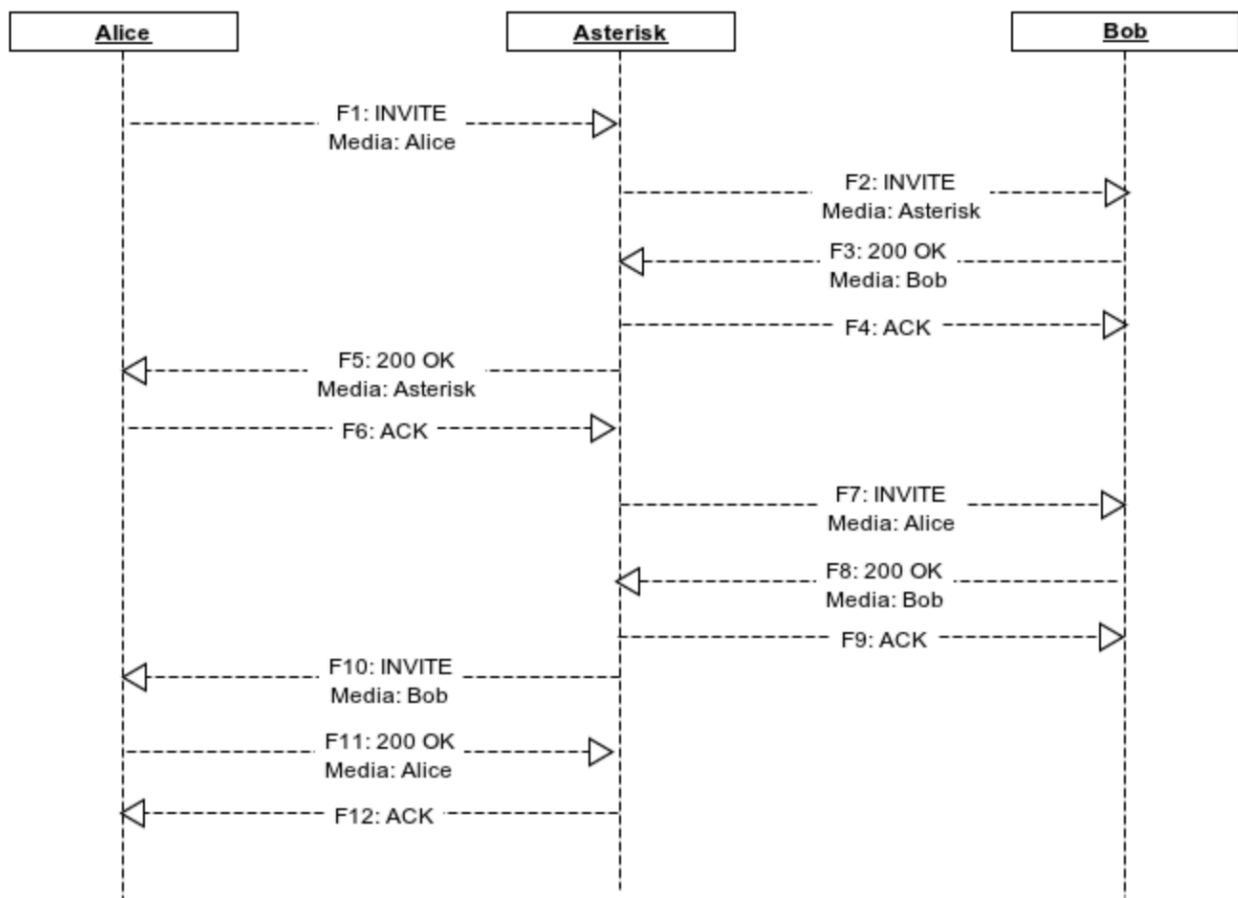
SIP ne transporte pas la voix lui-même! Il dit juste aux téléphones comment se connecter. La voix passe ensuite par un autre protocole (RTP).

**SIP est le protocole utilisé par Asterisk** pour faire transiter les appels entre les téléphones.

SIP est **déjà intégré** dans Asterisk. Asterisk supporte plusieurs protocoles pour gérer les appels VoIP, et **SIP est l'un des principaux**.

**SIP** Il définit la manière dont les appels sont initiés, dirigés et fermés.

Le trafic SIP typique pour un appel pourrait ressembler à ceci :



## RTP (Real-time Transport Protocol)

Le **RTP** (Real-time Transport Protocol) est un protocole clé dans la transmission de la voix et de la vidéo en temps réel sur Internet, notamment utilisé dans la VoIP (Voice over IP).

### Définition et Rôle du RTP

**RTP** est un protocole de transport conçu pour envoyer de l'audio et de la vidéo en temps réel sur un réseau IP. Il fonctionne généralement sur UDP (User Datagram Protocol), car UDP permet une transmission rapide sans attendre d'accusé de réception, ce qui est crucial pour éviter les latences dans la communication.

## Comment fonctionne RTP dans la VoIP ?

### 1. Encodage de la Voix

- Quand un utilisateur parle, sa voix est convertie en un **flux numérique** à l'aide d'un **codec** (comme G.711, G.729, Opus...).

### 2. Segmentation en Paquets RTP

- Ces données sont découpées en petits morceaux et **encapsulées** dans des paquets RTP.
- Chaque paquet RTP contient :
- Une partie audio/vidéo (données utiles)
- **Un en-tête RTP** (avec un numéro de séquence et un horodatage pour reconstituer correctement l'ordre des paquets à la réception).

### 3. Transport via UDP

- Les paquets RTP sont envoyés **via UDP** sur le réseau.
- UDP est utilisé car il est plus rapide que TCP et ne ralentit pas la transmission en cas de perte de paquets.

### 4. Réception et Reconstitution

- Le récepteur (ex : téléphone VoIP, softphone) reçoit les paquets RTP et utilise les **numéros de séquence et les horodatages** pour remettre en ordre les données audio.
- Le flux est ensuite **converti en son audible** et joué en temps réel.

## TLS (Transport Layer Security)

Pour sécuriser et chiffrer les appels transitant par le serveur Asterisk, on utilise généralement **TLS** (Transport Layer Security) pour sécuriser la signalisation SIP et **SRTP** (Secure Real-time Transport Protocol) pour chiffrer l'audio des appels.

**TLS** va sécuriser la communication entre les clients et le serveur Asterisk **en chiffrant les messages SIP**.

### Générer les certificats pour sécuriser Asterisk avec TLS

Asterisk a besoin de **certificats** pour chiffrer les appels avec **TLS**.

**Créer un dossier pour stocker les certificats:** `mkdir /etc/asterisk/keys`  
Cela crée un dossier où l'on va stocker les fichiers de chiffrement.



## Générer le certificat pour Asterisk (serveur)

On utilise l'outil `ast_tls_cert` pour créer les certificats. Tape cette commande :

**`./ast_tls_cert -C 192.168.1.100 -O "DataTrust" -d /etc/asterisk/keys -b 2048`**

```
bash: ./ast_tls_cert: NO SUCH FILE OR DIRECTORY
root@debian:/etc/asterisk/keys# find / -name ast_tls_cert 2>/dev/null
/usr/src/asterisk-18.26.1/contrib/scripts/ast_tls_cert
/usr/src/asterisk-22.2.0/contrib/scripts/ast_tls_cert
root@debian:/etc/asterisk/keys#
```

---

```
root@debian:/etc/asterisk/keys# ast_tls_cert -C 172.16.162.141 -O "silvia" -d /etc/asterisk/keys -b 2048

No config file specified, creating '/etc/asterisk/keys/tmp.cfg'
You can use this config file to create additional certs without
re-entering the information for the fields in the certificate
Creating CA key /etc/asterisk/keys/ca.key
Enter PEM pass phrase:

Creating CA key /etc/asterisk/keys/ca.key
[Enter PEM pass phrase:
[Verifying - Enter PEM pass phrase:
Creating CA certificate /etc/asterisk/keys/ca.crt
[Enter pass phrase for /etc/asterisk/keys/ca.key:
Creating certificate /etc/asterisk/keys/asterisk.key
Creating signing request /etc/asterisk/keys/asterisk.csr
Creating certificate /etc/asterisk/keys/asterisk.crt
Certificate request self-signature ok
subject=CN = 172.16.162.141, O = silvia
[Enter pass phrase for /etc/asterisk/keys/ca.key:
Combining key and crt into /etc/asterisk/keys/asterisk.pem
root@debian:/etc/asterisk/keys#
```

## Générer les certificats pour Alice et Bob (clients SIP)

Chaque utilisateur SIP (Alice et Bob) a besoin d'un certificat pour **se connecter en mode sécurisé**.

le script est situé ici : `/usr/src/asterisk-22.2.0/contrib/scripts/`  
`cd /usr/src/asterisk-22.2.0/contrib/scripts/`

## Exemple de création d'un certificat pour utilisateur Alice

```
./ast_tls_cert -m client -c /etc/asterisk/keys/ca.crt -k /etc/asterisk/keys/ca.key -C 172.16.162.141 -O "silvia" -d /etc/asterisk/keys -o alice -b 2048
```

```
root@debian:/usr/src/asterisk-22.2.0/contrib/scripts# ls /etc/asterisk/keys
alice.crt  alice.csr  alice.key  asterisk.crt  asterisk.csr  asterisk.key  asterisk.pem  ca.cfg  ca.crt  ca.key  tmp.cfg
root@debian:/usr/src/asterisk-22.2.0/contrib/scripts#
```

Pour vérifier la présence des fichiers:

### **ls /etc/asterisk/keys**

```
root@debian:/usr/src/asterisk-22.2.0/contrib/scripts# ls /etc/asterisk/keys
alice.crt  alice.csr  alice.key  asterisk.crt  asterisk.csr  asterisk.key  asterisk.pem  bob.crt  bob.csr  bob.key  bob.pem  ca.cfg  ca.crt  ca.key  tmp.cfg
root@debian:/usr/src/asterisk-22.2.0/contrib/scripts#
```

## Configurer Asterisk pour utiliser TLS

**protocol=tls** : Le protocole TLS doit être activé.

**bind=0.0.0.0:5061** : Le port 5061 est généralement utilisé pour TLS,

**cert\_file** et **priv\_key\_file** : Les chemins vers les certificats et la clé privée.

**method=sslv23** : Spécifie la méthode de chiffrement

### **[transport-tls]**

```
type=transport
protocol=tls
bind=0.0.0.0:5061
cert_file=/etc/asterisk/keys/asterisk.crt
priv_key_file=/etc/asterisk/keys/asterisk.key
method=sslv23
```

## Serveur vocal interactif (SVI)

Création d'un **serveur vocal interactif (SVI)**, aussi appelé **menu vocal** ou **serveur automatisé**, sur le serveur Asterisk.

Lorsqu'un appel arrive, l'appelant entend un message vocal lui demandant d'appuyer sur une touche pour accéder à un service spécifique

Pour le répondeur: \*97 messagerie  
Code d'accès:( mot de passe) 1234  
1 pour

Dans notre configuration **Asterisk**, les touches de téléphone sont associées à des actions spécifiques, notamment pour effectuer des appels, accéder à la messagerie vocale, ou naviguer dans des menus interactifs. Voici une explication de chaque section importante, ainsi que les touches correspondantes :

```
GNU nano 7.2 /etc/asterisk/voicemail.conf
; imapport                ; Overrides the global imapport setting
; imapflags                ; Overrides the global imapflags setting
;
;[imapvm]
;4324 => 7764,Ellis Redding,red@buxton.us,,imapuser=eredding|imappassword=g3tbusy|imapfolder=notinbox
;4325 => 2392,Andrew Dufresne,andy@dufresne.info,,imapuser=adufresne|imappassword=rockh@mmmer

[general]
format=wav49|gsm|wav
attach=no
emailbody=no
delete=no
maxmsg=100
maxsecs=180
minsecs=3

[main]
1001 => 1234,User One
1002 => 1234,User Two
```

```
GNU nano 7.2 /etc/asterisk/extensions.conf

[from-internal]
exten => 1001,1,NoOp()
exten => 1001,2,Set(CHANNEL(language)=fr)
exten => 1001,3,Dial(PJSIP/1001,20,tWr)
exten => 1001,4,NoOp(DIALSTATUS=${DIALSTATUS})
exten => 1001,5,GotoIf("${DIALSTATUS}" = "BUSY"?10:6)
exten => 1001,6,GotoIf("${DIALSTATUS}" = "NOANSWER"?15:7)
exten => 1001,7,GotoIf("${DIALSTATUS}" = "CHANUNAVAIL"?15:8)
exten => 1001,8,Playback(vm-nobodyavail)
exten => 1001,9,Goto(15)
exten => 1001,10,Playback(vm-isonphone)
exten => 1001,11,VoiceMail(1001@main,b)
exten => 1001,12,Hangup()
exten => 1001,15,Answer()
exten => 1001,16,Wait(1)
exten => 1001,17,Playback(vm-nobodyavail)
exten => 1001,18,VoiceMail(1001@main,u)
exten => 1001,19,Hangup()
```

```

[from-internal]
exten => 1001,1,Dial(SIP/1001,20)
exten => 1001,n,VoiceMail(1001@default,u)

exten => 1002,1,Dial(SIP/1002,20)
exten => 1002,n,VoiceMail(1002@default,u)

exten => *97,1,Answer()
exten => *97,2,Wait(1)
exten => *97,3,VoiceMailMain(${CALLERID(num)}@main)
exten => *97,4,Hangup()

[general]
[menu-principal]
exten => s,1,Answer()
exten => s,n,Wait(1)
exten => s,n,Background(bienvenue)
exten => s,n,WaitExten(5)

exten => 1,1,Goto(service-compte,s,1)
exten => 2,1,Goto(service-rh,s,1)
exten => t,1,Hangup()
exten => i,1,Playback(option-invalide)
exten => i,n,Goto(menu-principal,s,1)

```

## Extensions 1001 et 1002 (appels internes)

- Extension 1001 (numéro spécifique qui est associé à un utilisateur):
  - Appel de l'extension 1001 :  
L'extension 1001 est configurée pour appeler un autre utilisateur SIP/1001 pendant 20 secondes.
  - Messagerie vocale :  
Si l'appel est infructueux (occupé, sans réponse ou canal non disponible), le système redirige l'appelant vers la messagerie vocale avec la commande `VoiceMail(1001@default,u)`.

## Messagerie vocale (écouter la messagerie)

- \*97 :  
Cette extension permet à l'utilisateur d'accéder à sa messagerie vocale personnelle pour écouter ses messages.D

## L'utilisation d'un softphone

Pour notre projet, nous avons utilisé l'application **PortSIP**

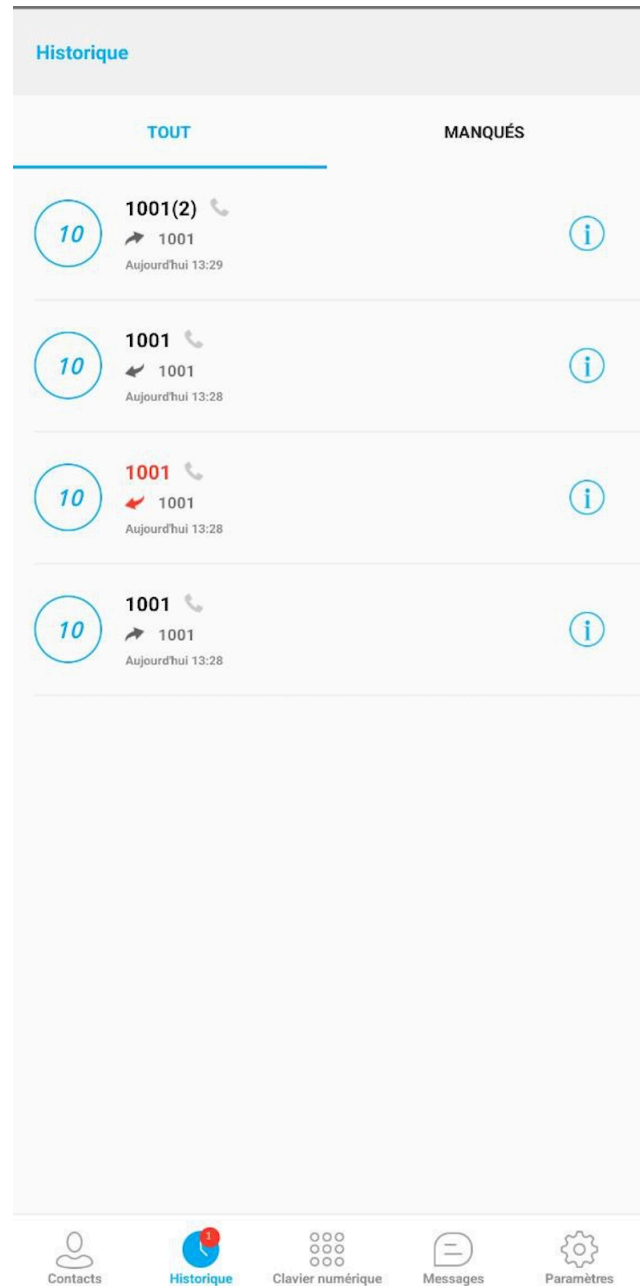
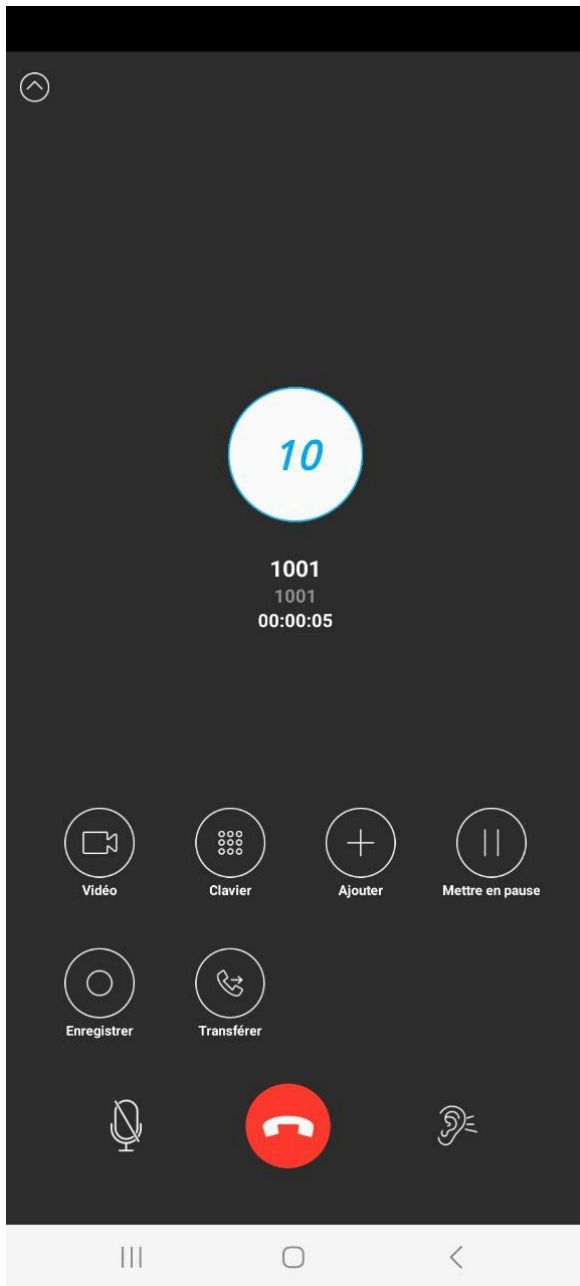
Mais d'autres alternatives existent:

1. **Zoiper** : Un softphone léger et facile à utiliser, disponible sur plusieurs plateformes (Windows, macOS, Linux, Android, iOS). Il prend en charge plusieurs protocoles, dont SIP et IAX.
2. **Linphone** : Un softphone open-source qui prend en charge les appels audio et vidéo via SIP. Disponible sur plusieurs plateformes, y compris Android, iOS, Windows et Linux.
3. **X-Lite** : Un softphone gratuit de la société CounterPath, très populaire pour sa simplicité et sa prise en charge des appels VoIP via SIP. Il est disponible pour Windows et macOS.
4. **MicroSIP** : Une solution simple et légère pour Windows, qui permet de passer des appels VoIP via SIP avec une faible utilisation des ressources système.
5. **CSipSimple** : Un softphone open-source pour Android, compatible avec le protocole SIP. Il est connu pour sa flexibilité et sa personnalisation.
6. **Bria** : Un softphone payant de CounterPath qui offre une interface professionnelle et des fonctionnalités avancées pour les entreprises. Il est disponible sur Windows, macOS, Android et iOS.

Tous ces softphones utilisent le protocole SIP pour établir des appels VoIP et peuvent être configurés avec des serveurs comme Asterisk.

Le softphone peut être installé sur un ordinateur, une tablette ou un smartphone. Il suffit que l'appareil soit, à minima, doté d'un microphone et d'un haut-parleur. Le softphone permet de recevoir ou d'émettre des appels aussi bien depuis un numéro fixe que depuis un numéro mobile.

# Phase de test



Réalisation  
d'appels entre deux smartphones via l'application PortSIP, avec un **résultat positif**

## **Conclusions**

Dans un monde de plus en plus connecté et axé sur la mobilité, les solutions de communication flexibles se sont imposées comme des outils incontournables.

Ce projet a permis de réaliser l'installation et la configuration d'Asterisk, un serveur de téléphonie open-source, sur une machine virtuelle Debian.

Asterisk a été configuré pour gérer les appels VoIP, en mettant en place des extensions et des services de messagerie vocale.

L'intégration de softphones tels que PortSIP a été réalisée pour simuler des appels entre différents utilisateurs, permettant ainsi de tester la fonctionnalité du système en situation réelle.

Nous avons configuré différents éléments du système, tels que les extensions, les menus interactifs et la messagerie vocale.

Ce projet a permis d'acquérir une compréhension approfondie des principes de base de la téléphonie VoIP et de la gestion des communications à travers Asterisk.

