



Université Sorbonne Paris Nord : IUT Villetaneuse
Département Réseaux et Télécommunication
Dr. Yamina Amzal et Mohamed Amine Ouamri

SAE : Concevoir un réseau informatique adapté au
multimédia

Pour Information :

Vous devez Exposer la démonstration de votre installation 4h + 2h classique
2 étudiants par groupe avec des captures sur la bonne réalisation du projet.

- ✓ 15 minutes de présentation
- ✓ 10 min question

Objectif

Dans le cadre de cette SAE, vous allez mettre en place un serveur Asterisk, en exploitant habilement les synergies entre deux technologies essentielles : la VoIP (Voice over Internet Protocol) et le WebRTC (Web Real-Time Communication). Ce déploiement stratégique est spécifiquement conçu pour répondre aux besoins de deux clients enthousiastes distancié. Grâce à cette approche novatrice, vous allez établir une connectivité vocale et audio en temps réel, surmontant les frontières géographiques et permettant une collaboration fluide et immersive entre les parties prenantes situées dans ces deux régions distinctes.

Voici donc vos différentes missions :

Mission 1 : Installer et prendre en main l'IPBX Asterisk :

- ✓ Vous devez installer et configurer l'IPBX Asterisk dans différents environnement (machine virtuelle, container docker, PC).
- ✓ Configurer les flux entrants et sortants
- ✓ Configuration des SIP Phones et des Softphones.

Mission 2 : Combiner les deux technologies WebRTC et VoIP

- ✓ Produire un document synthèse des recherches effectuées sur les technologies
- ✓ Mettre en œuvre la communication entre l'application WebRTC et l'IPBX Asterisk
- ✓ Produire un rapport, mettant en évidence les bilans des résultats obtenus

1. Installer et prendre en main l'IPBX Asterisk

Actuellement, tout le monde utilise la VoIP (Voice over IP) ou de la ToIP (Telephony over IP). Depuis plusieurs années, ce secteur est en pleine croissance grâce à deux composantes essentielles : les utilisateurs (grand public comme entreprises) qui sont en majorité séduits et les constructeurs/éditeurs qui investissent énormément pour proposer des offres les plus complètes possibles.

En 2002, le projet Asterisk sort au grand jour et fait son entrée dans un marché encore naissant. C'est un PBX (Private Branch eXchange) logiciel qui propose des fonctionnalités avancées pour une somme dérisoire car la (bonne) surprise est que sa licence GPL (donc projet libre et open-source). D'abord utilisé plus ou moins expérimentalement, il commence à convaincre peu à peu les entreprises de toute taille. Asterisk est un serveur téléphonique IP (PBX-IP) open source capable de concurrencer des systèmes commerciaux tels que les Call Manager de Cisco System.

1.1. Fonctionnement

Asterisk fournit tous les services de base d'un PABX comme la connexion des postes entre eux (qu'ils

soient locaux ou distants), messagerie unifiée, services Web intégrés (ex: annuaire, gestion salle de conférence, etc.), service de répondeur interactif (IVR), musique d'attente, interconnexion avec le réseau téléphonique public, etc. Asterisk est basé comme le Cisco Call Manager sur le principe de canaux (Channels), de plan de numérotation (Dial Plan) et de contextes (search spaces pour le Cisco Call Manager). Quand un appel arrive sur canal, par exemple le canal SIP, le but du PABX - grâce au plan de numérotation - est de trouver le canal de sortie qui peut être le même canal SIP dans le cas d'un appel en VoIP à l'intérieur du bâtiment, ce canal de sortie peut également être un des autres types de canaux géré par Asterisk. Les contextes servent à réduire (ou augmenter) les possibilités de sortie d'un appel. Cela peut par exemple servir pour autoriser les appels à l'international pour certains utilisateurs seulement, pour créer des services d'IVR (lors d'un appel sur un numéro, on fait appel à un contexte qui déroule le script du répondeur automatique, etc.).

Asterisk utilise de multiples canaux d'entrée/sortie qui peuvent être de type IP comme MGCP, IAX, SIP, H.323, skinny, etc. ou de type téléphonie classique avec entre autre Zap (FXS et FXO), ISDN (BRI et PRI).

1.2. Installation et configuration

Dans cette sous-section, vous allez installer et configurer l'IPBX Asterisk dans différents environnement (machine virtuelle, container docker, micro PC). Pour commencer, vous devez commencer par télécharger Asterisk 18.5.1.

`wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-18.5.1.tar.gz`

```
kendy@kendy:~$ wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-18.15.1.tar.gz
--2023-03-03 11:05:33-- http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-18.15.1.tar.gz
Résolution de downloads.asterisk.org (downloads.asterisk.org)... 170.249.154.172
Connexion à downloads.asterisk.org (downloads.asterisk.org)|170.249.154.172|:80... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
```

```
asterisk-18.15.1.tar.gz 100%[=====]
2023-03-03 11:05:48 (1,79 MB/s) - 'asterisk-18.15.1.tar.gz' enregistré [28216621/28216621]
```

Decompression du dossier Asterisk - `tar -xzf asterisk-18.5.1.tar.gz`

```
kendy@kendy:~$ tar xzvf asterisk*
asterisk-18.15.1/
asterisk-18.15.1/.cleancount
asterisk-18.15.1/.gitignore
asterisk-18.15.1/.gitreview
asterisk-18.15.1/.lastclean
asterisk-18.15.1/.version
asterisk-18.15.1/BSDmakefile
asterisk-18.15.1/BUGS
asterisk-18.15.1/CHANGES
asterisk-18.15.1/COPYING
asterisk-18.15.1/CREDITS
asterisk-18.15.1/ChangeLog
asterisk-18.15.1/LICENSE
asterisk-18.15.1/Makefile
asterisk-18.15.1/Makefile.moddir_rules
asterisk-18.15.1/Makefile.rules
asterisk-18.15.1/README-SERIOUSLY.bestpractices.md
asterisk-18.15.1/README-addons.txt
asterisk-18.15.1/README.md
asterisk-18.15.1/UPGRADE.txt
asterisk-18.15.1/Zaptel-to-DAHDI.txt
asterisk-18.15.1/addons/
asterisk-18.15.1/addons/.gitignore
asterisk-18.15.1/addons/Makefile
asterisk-18.15.1/addons/app_mysql.c
asterisk-18.15.1/addons/cdr_mysql.c
asterisk-18.15.1/addons/chan_mobile.c
asterisk-18.15.1/addons/chan_ooh323.c
asterisk-18.15.1/addons/chan_ooh323.h
asterisk-18.15.1/addons/format_mp3.c
```

Rentrer dans le dossier Asterisk:- `cd asterisk 18.15.1/`

```
kendy@kendy:~$ cd asterisk-18.15.1/
kendy@kendy:~/asterisk-18.15.1$
```

Compilation du fichier (configure) : `./configure`

```
root@kendy:/usr/src/asterisk/asterisk-18.15.1# ./configure --with-pjproject-bundled
```



```

+---- 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  +
+-----+

```

Mettre à jour les librairies : **-ldconfig**

```

[root@kendy:/usr/src/asterisk/asterisk-18.15.1# ldconfig
[root@kendy:/usr/src/asterisk/asterisk-18.15.1# ldd /usr/sbin/as
aspell-autobuildhash asterisk astgenkey astversion
[root@kendy:/usr/src/asterisk/asterisk-18.15.1# ldd /usr/sbin/asterisk
linux-vdso.so.1 (0x00007fffd6d2be000)
libasteriskssl.so.1 => /lib/libasteriskssl.so.1 (0x00007fafdcabe000)
libasteriskpj.so.2 => /lib/libasteriskpj.so.2 (0x00007fafdc839000)
libxml2.so.2 => /lib/x86_64-linux-gnu/libxml2.so.2 (0x00007fafdc657000)

libxslt.so.1 => /lib/x86_64-linux-gnu/libxslt.so.1 (0x00007fafdc615000)
libsqlite3.so.0 => /lib/x86_64-linux-gnu/libsqlite3.so.0 (0x00007fafdc4c8000)
libssl.so.3 => /lib/x86_64-linux-gnu/libssl.so.3 (0x00007fafdc424000)
libcrypto.so.3 => /lib/x86_64-linux-gnu/libcrypto.so.3 (0x00007fafdbbfd000)
libjansson.so.4 => /lib/x86_64-linux-gnu/libjansson.so.4 (0x00007fafdbfcf000)
liburiparser.so.1 => /lib/x86_64-linux-gnu/liburiparser.so.1 (0x00007fafdbfb8000)
libuuid.so.1 => /lib/x86_64-linux-gnu/libuuid.so.1 (0x00007fafdbfaf000)
libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007fafdbf75000)
libcap.so.2 => /lib/x86_64-linux-gnu/libcap.so.2 (0x00007fafdbf6a000)
libbfd-2.38-system.so => /lib/x86_64-linux-gnu/libbfd-2.38-system.so (0x00007fafdbdf0000)
libedit.so.2 => /lib/x86_64-linux-gnu/libedit.so.2 (0x00007fafdbdb6000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fafdbccf000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fafdbaa7000)
libcuc.so.70 => /lib/x86_64-linux-gnu/libcuc.so.70 (0x00007fafdb8ac000)

```

Lancement du serveur Asterisk : **- asterisk -rvvvv**

```

[root@kendy:/usr/src/asterisk/asterisk-18.15.1# asterisk -rvvvv
Asterisk 18.15.1, 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 18.15.1 currently running on kendy (pid = 44376)
[kendy*CLI>

```

Une fois la configuration d'Asterisk est réalisée, vous devez configurer flux entrants et sortants. Vous commencez par configurer le fichier **sip.conf** en utilisant **nano /etc/asterisk/sip.conf**

```
root@kendy:/home/kendy# nano /etc/asterisk/sip.conf
```

La configuration générale du fichier **sip.conf** est la suivante :

```
[general]
context=internal
allowguest=no
allowoverlap=no
bindport=5060
bindaddr=0.0.0.0
srlookup=no
disallow=all
allow=ulaw
alwaysauthreject=yes
canreinvite=no
nat=yes
session-timers=refuse
localnet=192.168.0.0/255.255.255.0
```

Configuration des différents utilisateurs

```
[friend]
type=friend
context=from-internal
host=dynamic
secret=password
disallow=all
allow=ulaw
```

```
[friend]
type=friend
context=from-internal
host=dynamic
secret=5002
disallow=all
allow=ulaw
```

```
[friend]
type=friend
context=from-internal
host=dynamic
secret=password
disallow=all
allow=ulaw
```

Configuration du fichier **extensions.conf** nano /etc/asterisk/extensions.conf

```
root@kendy:/home/kendy# nano /etc/asterisk/extensions.conf
```

Information mis dans le fichier **extensions.conf**

```
[from-internal]
exten=>5001,1,Dial(SIP/5001,20)
exten=>5002,1,Dial(SIP/5002,20)
exten=>5003,1,Dial(SIP/5003,20)

exten = 5000,1,Answer()
same = n,Wait(1)
same = n,Playback(hello-word)
same = n,Hangup()
```

Pour finir, vous configurez le fichier **voicemail.conf** -nano /etc/asterisk/voicemail.conf

```
root@kendy:/home/kendy# nano /etc/asterisk/voicemail.conf
```

Les informations à mettre dans le fichier sont les suivants

```
[main]
5001 => 5001

5002 => 5002

5003 => 5003
```

Configurer des SIP Phones et des Softphones.

Grace à la commande **asterisk-r** : vous avez accès a liste des utilisateurs installés : **sip show peers**

```
kendy*CLI> sip show peers
Name/username      Host                      Dyn Forcerport
Comedia    ACL Port    Status    Description
5001
Yes          0          Unmonitored (Unspecified)      D Yes
5002/5002
Yes          56858      Unmonitored 10.10.11.80      D Yes
5003
Yes          0          Unmonitored (Unspecified)      D Yes
3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 1 online, 2 offline]
```

Dans la suite de cette première mission installé deux téléphone IP pour pourvoir les connecter au serveur Asterisk voir la figure ci-dessous

	<p>Nom : 5001</p> <p>IP : 192.168.1.91</p> <p>IP du serveur Asterisk : 192.168.1.96</p> <p>Port : 5060</p> <p>Mot-de-passe : password</p>		<p>Nom : 5003</p> <p>IP : 192.168.1.94</p> <p>IP du serveur Asterisk : 192.168.1.96</p> <p>Port : 5060</p> <p>Mot-de-passe : password</p>
---	---	--	---

Depuis le serveur Asterisk démarré, on peut remarquer que les postes sont prêts à l'utilisation

```
kendy*CLI> sip show peers
Name/username      Host                      Dyn Forcerport
Comedia    ACL Port    Status    Description
5001/5001
Yes          5060      Unmonitored 192.168.1.91      D Yes
5002/5002
Yes          56858      Unmonitored 10.10.11.80      D Yes
5003/5003
Yes          5060      Unmonitored 192.168.1.94      D Yes
3 sip peers [Monitored: 0 online, 0 offline Unmonitored: 3 online, 0 offline]
```

2. Combiner les deux technologies WebRTC et VoIP

WebRTC est un effort pour apporter une API définie aux développeurs JavaScript qui leur permet de s'aventurer dans le monde des communications en temps réel. Il peut s'agir d'un système d'appel en un clic ou d'un "softphone", les deux étant livrés sous forme de page Web. Aucun plug-in n'est requis et comme il s'agit d'une spécification définie, il peut être utilisé sur différents navigateurs lorsqu'ils sont pris en charge. Asterisk prend en charge WebRTC depuis la version 11. Un module `res_http_websocket` a été créé qui permet aux développeurs JavaScript d'interagir et de communiquer avec Asterisk. La prise en charge de WebSocket en tant que transport a été ajoutée à `chan_sip` pour permettre à SIP d'être utilisé comme protocole de signalisation. Le support ICE, STUN et TURN a été ajouté à `res_rtp_asterisk` pour permettre aux clients derrière NAT de mieux communiquer avec Asterisk.

Le support SRTP a été ajouté dans une version précédente, mais c'est également une exigence de WebRTC. Le support sécurisé est une exigence de WebRTC et, par conséquent, SRTP doit être disponible.

Pour qu'Asterisk puisse créer le support STP, la bibliothèque `libsrtp` et les en-têtes de développement doivent être disponibles. Cela peut être installé à l'aide du système de gestion des packages de la distribution ou à partir de la source. Si vous ne le faites pas, les offres médias seront rejetées.

WebRTC utilise des `MediaStreamTrack` objets nus pour chaque piste partagée d'un homologue à un autre, sans conteneur ni même `MediaStream` associé aux pistes. Les codes pouvant se trouver dans ces pistes ne sont pas prescrits par la spécification WebRTC. Cependant, la RFC 7742 spécifie que tous les navigateurs compatibles WebRTC doivent prendre en charge le profil de base contraint de VP8 et H.264 pour la vidéo, et la RFC 7874 spécifie que les navigateurs doivent au moins prendre en charge le codec Opus ainsi que les PCMA et PCMU de G.711. Le G.711 est une norme de compression audio de l'UIT T qui définit les codages PCM-U et PCM-A, s'appuyant sur les lois de quantification A (Europe, Afrique) ou U (Amérique du Nord, Japon).

Dans cette partie, vous allez réinstaller un serveur Asterisk sur des machines comme celui déjà vu précédemment mais en modifiant certains paramètres et en ajoutant d'autre.

Serveur Asterisk lancé :

Vous crée 6 utilisateurs (4 postes de consultations et 2 téléphone portable) pour les mettre en relation.

```
Connected to Asterisk 18.15.1 currently running on raspberry (pid = 5922)
raspberrypi*CLI> sip show peers
Name/username      Host                               Dyn Forcerport Comedia  ACL Port  Status
5001/5001           (Unspecified)                   D Yes      Yes      0         UNKNOWN
User1               (Unspecified)                   D Yes      Yes      0         UNKNOWN
User2/kg4e9qg0      192.168.1.22                    D Yes      Yes      51392     OK (5 ms)
User3               (Unspecified)                   D Yes      Yes      0         UNKNOWN
User4               (Unspecified)                   D Yes      Yes      0         UNKNOWN
User5               (Unspecified)                   D Yes      Yes      0         UNKNOWN
6 sip peers [Monitored: 1 online, 5 offline Unmonitored: 0 online, 0 offline]
```

Pour ce serveur, vous devez modifier 5 fichiers (2 configurations possibles → SIP/PJSIP)

Le fichier [http.conf](#) est un fichier de configuration d'Asterisk qui permet de configurer le module HTTP. Ce module permet d'accéder à une interface web d'administration d'Asterisk, appelée "Asterisk HTTP Manager Interface" (AMI), ainsi que d'autres fonctionnalités liées à http ([nano /etc/asterisk/http.conf](#))

```
[general]
enabled=no ; HTTP
tlsenable=yes ; HTTPS
tlsbindaddr=0.0.0.0:443
tlscertfile=/home/kendy/certs/raspberrypi.crt
tlsprivatekey=/home/kendy/certs/raspberrypi.key
enablestatic=yes
sessionlimit=1000
redirect=/ /static/index.html
```

La configuration pjsip permet de faire la passerelle entre les communications des téléphones SIP et la plateforme WebRTC. Le fichier [pjsip.conf](#) est essentiel pour configurer le module PJSIP d'Asterisk et définir le comportement des appels SIP. Il est important de comprendre la syntaxe et les options de configuration disponibles dans le fichier [pjsip.conf](#) pour configurer correctement les points d'extrémité, les informations d'authentification, les transports réseau et d'autres paramètres liés aux appels SIP. ([nano /etc/asterisk/pjsip.conf](#))

Ci-joint un exemple de configuration pour un Raspberry pi.

→ Paramètres générales

```
[global]
max_forwards=70
user_agent=Raspberry Pi BPX
default_real=raspberrypi.local
keep_alive_interval=300
```

→ Paramètres des transports

```
; == Transports

[udp_transport]
type=transport
protocol=udp
bind=0.0.0.0
tos=af42
cos=3

[wss_transport]
type=transport
protocol=wss
bind=0.0.0.0

[tcp_transport]
type=transport
protocol=tcp
bind=0.0.0.0

[tls_transport]
type=transport
protocol=tls
bind=0.0.0.0
cert_file=/home/kendy/certs/raspberrypi.crt
priv_key_file=/home/kendy/certs/raspberrypi.key
cipher=ADH-AES256-SHA,ADH-AES128-SHA
method=tlsv1
```

→ Paramètres des protocoles

```
; == ACL

[acl] ; Operates on all pjsip traffic (can also be in acl.conf)
type=acl
deny=0.0.0.0/0.0.0.0
permit=10.0.0.0/255.0.0.0
permit=172.16.0.0/255.240.0.0
permit=192.168.0.0/255.255.0.0

[single_aor]()
max_contacts=1
qualify_frequency=120
remove_existing=yes

[userpass_auth]()
auth_type=userpass

[phone_endpoint]()
allow=ulaw,alaw,g722,gsm,vp9,vp8,h264

[webrtc_endpoint]()
transport=wss_transport
allow=opus,ulaw,vp9,vp8,h264
; webrtc=yes
use_avpf=yes
media_encryption=dtls
dtls_verify=fingerprint
dtls_setup=actpass
ice_support=yes
media_use_received_transport=yes
rtcp_mux=yes
dtls_cert_file=/home/kendy/certs/raspberrypi.crt
dtls_private_key=/home/kendy/certs/raspberrypi.key
dtls_ca_file=/home/kendy/ca/InnovateAsterisk-Root-CA.crt

[basic_endpoint]()
moh_suggest=default
context=from-extensions
inband_progress=no
rtp_timeout=120
message_context=textmessages
allow_subscribe=yes
subscribe_context=subscriptions
direct_media=no
dtmf_mode=rfc4733
device_state_busy_at=1
disallow=all
```

→ Paramètres des utilisateurs

```
[User1](basic_endpoint,webrtc_endpoint)
type=endpoint
callerid="JEROME KENDY" <100>
auth=User1
aors=User1
[User1](single_aor)
type=aor
mailboxes=User1@default
[User1](userpass_auth)
type=auth
username=User1
password=1234

[User2](basic_endpoint,webrtc_endpoint)
type=endpoint
callerid="User Two" <200>
auth=User2
aors=User2
[User2](single_aor)
type=aor
[User2](userpass_auth)
type=auth
username=User2
password=1234

[User3](basic_endpoint,phone_endpoint)
type=endpoint
callerid="User 3" <300>
auth=User3
aors=User3
[User3](single_aor)
type=aor
[User3](userpass_auth)
type=auth
username=User3
password=1234
```

Le fichier extensions.conf est un élément clé de la configuration d'Asterisk et joue un rôle central dans la personnalisation du système de téléphonie. Il est important de comprendre la syntaxe et les

fonctionnalités disponibles dans le fichier extensions.conf pour configurer correctement le plan de numérotation et le routage des appels selon vos besoins spécifiques. **nano /etc/asterisk/extensions.conf**

→ Paramètres générales

```
[general]
static=yes
writeprotect=no
priorityjumping=no
autofallthrough=no

[globals]
ATTENDED_TRANSFER_COMPLETE_SOUND=beep
```

→ Paramètres des messages écrits

```
[textmessages]
exten => 100,1,Gosub(send-text,s,1,(User1))
exten => 200,1,Gosub(send-text,s,1,(User2))
exten => 300,1,Gosub(send-text,s,1,(User3))
exten => 400,1,Gosub(send-text,s,1,(User4))
exten => 500,1,Gosub(send-text,s,1,(User5))
exten => 5001,1,Gosub(send-text,s,1,(5001))
exten => e,1,Hangup()
```

→ Paramètres des abonnements

```
[subscriptions]
exten => 100, hint, PJSIP/User1
exten => 200, hint, PJSIP/User2
exten => 300, hint, PJSIP/User3
exten => 400, hint, PJSIP/User4
exten => 500, hint, PJSIP/User5
exten => 5001, hint, PJSIP/5001
```

→ Paramètres des extensions

```
[from-extensions]
; Feature Codes:
exten => *65,1,Gosub(moh,s,1)
; Extensions
exten => 100,1,Gosub(dial-extension,s,1,(User1))
exten => 200,1,Gosub(dial-extension,s,1,(User2))
exten => 300,1,Gosub(dial-extension,s,1,(User3))
exten => 400,1,Gosub(dial-extension,s,1,(User4))
exten => 500,1,Gosub(dial-extension,s,1,(User5))
exten => 5001,1,Gosub(dial-extension,s,1,(5001))
; Anything else, Hangup
exten => _[+*0-9]..1,NoOp(You called: ${EXTEN})
exten => _[+*0-9]..n,Hangup(1)
exten => e,1,Hangup()
```

→ Paramètres de la musique d'attentes

```
[moh]
exten => s,1,NoOp(Music On Hold)
exten => s,n,Ringing()
exten => s,n,Wait(2)
exten => s,n,Answer()
exten => s,n,Wait(1)
exten => s,n,MusicOnHold()
```

→ Paramètres des extensions de numéros

```
[dial-extension]
exten => s,1,NoOp(Calling: ${ARG1})
;exten => s,n,Set(JITTERBUFFER(adaptive)=default)
exten => s,n,Dial(PJSIP/${ARG1},30)
exten => s,n,Hangup()
exten => e,1,Hangup()
```

→ Paramètres des textes envoyés

```
[send-text]
exten => s,1,NoOp[Sending Text To: ${ARG1} From: ${MESSAGE(from)}]
exten => s,n,Set(PEER=${CUT(CUT(CUT(MESSAGE(from),@,1),<,2),:,2)})
exten => s,n,Set(FROM=${SHELL(asterisk -rx 'pjsip show peer ${PEER}' | grep 'Callerid' | cut -d':' -f2- | sed 's/^\s*//' | tr -d '\n')})
exten => s,n,Set(CALLERID_NUM=${CUT(CUT(FROM,>,1),<,2)})
exten => s,n,Set(FROM_SIP=${STRREPLACE(MESSAGE(from),<sip:${PEER}@,<sip:${CALLERID_NUM}@)})
exten => s,n,MessageSend(pjsip:${ARG1},${FROM_SIP})
exten => s,n,Hangup()
```

Quant à la configuration SIP elle communique seulement de plateformes à plateformes sauf exception où le téléphone IP peut communiquer avec une plateforme WebRTC en ayant les bons protocoles actifs.

Le fichier `sip.conf` est un fichier de configuration utilisé par Asterisk pour configurer le protocole SIP (Session Initiation Protocol). SIP est un protocole de signalisation largement utilisé dans les systèmes de téléphonie IP pour établir, modifier et terminer les sessions de communication.

→ Paramètres générales

```
[general]
accept_outofcall_message=yes
auth_message_requests=no
outofcall_message_context=textmessages
udpbindaddr=0.0.0.0:5060

context=from-outside
useragent=Raspberry Pi BPX
realm=raspberrypi.local
srvlookup=no
mohsuggest=default
parkinglot=default
allowguest=yes
alwaysauthreject=yes
videosupport=no
maxcallbitrate=5120
ignoreregexpire=no

allowssubscribe=yes
notifyhold=yes
notifyringing=yes
callcounter=yes

progressinband=yes ; yes|no|never

tos_sip=af42
tos_audio=ef
cos_sip=3
cos_audio=5

rtptimeout=120
rtpkeepalive=60

jbenable=yes
jbforce=no
jbmaxsize=200
jbsyncthreshold=1000
jbimpl=fixed
jblog=no

tcpenable=no
tlseable=no
websocket_enabled=yes
encryption=no

deny=0.0.0.0/0.0.0.0
permit=10.0.0.0/255.0.0.0
permit=172.16.0.0/255.240.0.0
permit=192.168.0.0/255.255.0.0
```

→ Paramètres des protocoles

```
[basic]()
type=friend
qualify=yes
context=from-extensions
subscribecontext=subscriptions
host=dynamic
directmedia=no
nat=force_rport,comedia
dtmfmode=rfc2833
disallow=all
videosupport=yes

[phones]()
transport=udp
allow=ulaw,alaw,g722,gsm,vp9,vp8,h264

[webrtc]()
transport=wss
allow=opus,ulaw,vp9,vp8,h264,g722

encryption=yes
avpf=yes
force_avp=yes
icesupport=yes
rtcp_mux=yes
dtlsenable=yes
dtlsverify=fingerprint
dtlscertfile=/home/kendy/certs/raspberrypi.pem
dtlscafile=/home/kendy/ca/InnovateAsterisk-Root-CA.crt
dtlssetup=actpass
```

→ Paramètres des utilisateurs

```
[User1](basic,webrtc)
callerid="JEROME KENDY" <100>
secret=1234

[User2](basic,webrtc)
callerid="User 2" <200>
secret=1234

[User3](basic,phones)
callerid="User 3" <300>
secret=1234

[User4](basic,webrtc)
callerid="User 4" <400>
secret=1234

[User5](basic,webrtc)
callerid="User 5" <500>
secret=1234

[5001](basic,phones)
callrid="5001" <5001>
secret=password
```

extensions.conf (configuré pour le fichier sip.conf) **nano /etc/asterisk/extensions.conf**

→ Paramètres des abonnements

```
[subscriptions]
exten => 100, hint, SIP/User1
exten => 200, hint, SIP/User2
exten => 300, hint, SIP/User3
exten => 400, hint, SIP/User4
exten => 500, hint, SIP/User5
exten => 5001, hint, SIP/5001
```

→ Paramètres des textes envoyés

```
[send-text]
exten => s,1,NoOp(Sending Text To: ${ARG1})
exten => s,n,Set(PEER=${CUT(CUT(CUT(MESSAGE(from),@,1),<,2),:,2)})
exten => s,n,Set(FROM=${SHELL(asterisk -rx 'sip show peer ${PEER}' | grep 'Callerid' | cut -d':' -f2- | sed 's/^\\ *//' | tr -d '\\n')})
exten => s,n,Set(CALLERID_NUM=${CUT(CUT(FROM,>,1),<,2)})
exten => s,n,Set(FROM_SIP=${STRREPLACE(MESSAGE(from),< sip:${PEER}@,< sip:${CALLERID_NUM}@)})
exten => s,n,MessageSend(sip:${ARG1},${FROM_SIP})
exten => s,n,Hangup()
```

Attention : Pour que les configurations (PJSIP/SIP) fonctionnent il ne doivent être utilisées en même temps, c'est pour cela qu'il existe un fichier nommé modules.conf.

modules.conf (configuré pour le fichier pjsip.conf) **nano /etc/asterisk/modules.conf**

→ Paramètre des modules

```
[modules]

autoload=yes

preload => res_odbc.so
preload => res_config_odbc.so
```

→ Paramètres PJSIP

```
; ---- Desactiver sip.conf ----

noload => chan_sip.so
noload => res_pjsip_config_wizzard.so
load = chan_pjsip.so
load = chan_bridge_media.so
```

modules.conf (configuré pour le fichier sip.conf) **nano /etc/asterisk/modules.conf**

→ Paramètre des modules

```
[modules]

autoload=yes

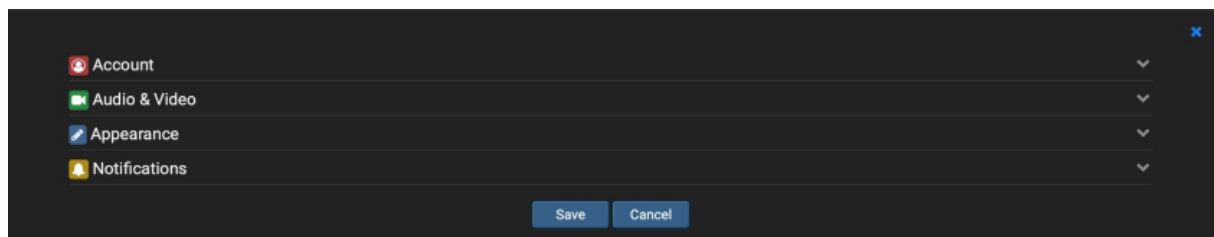
preload => res_odbc.so
preload => res_config_odbc.so
```

→ Paramètres SIP

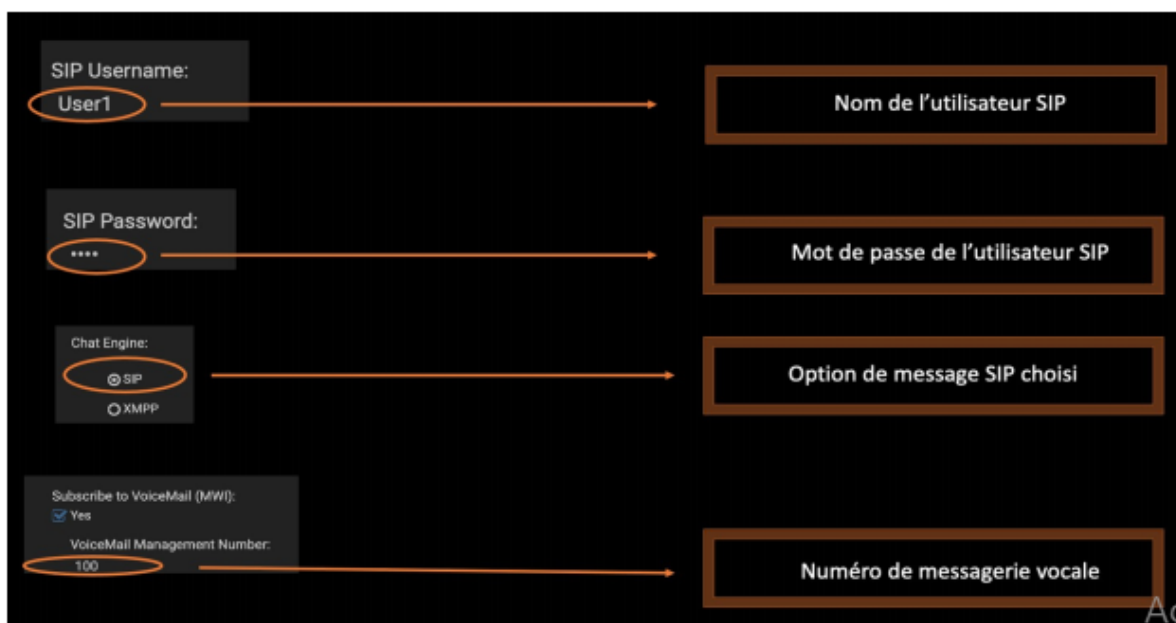
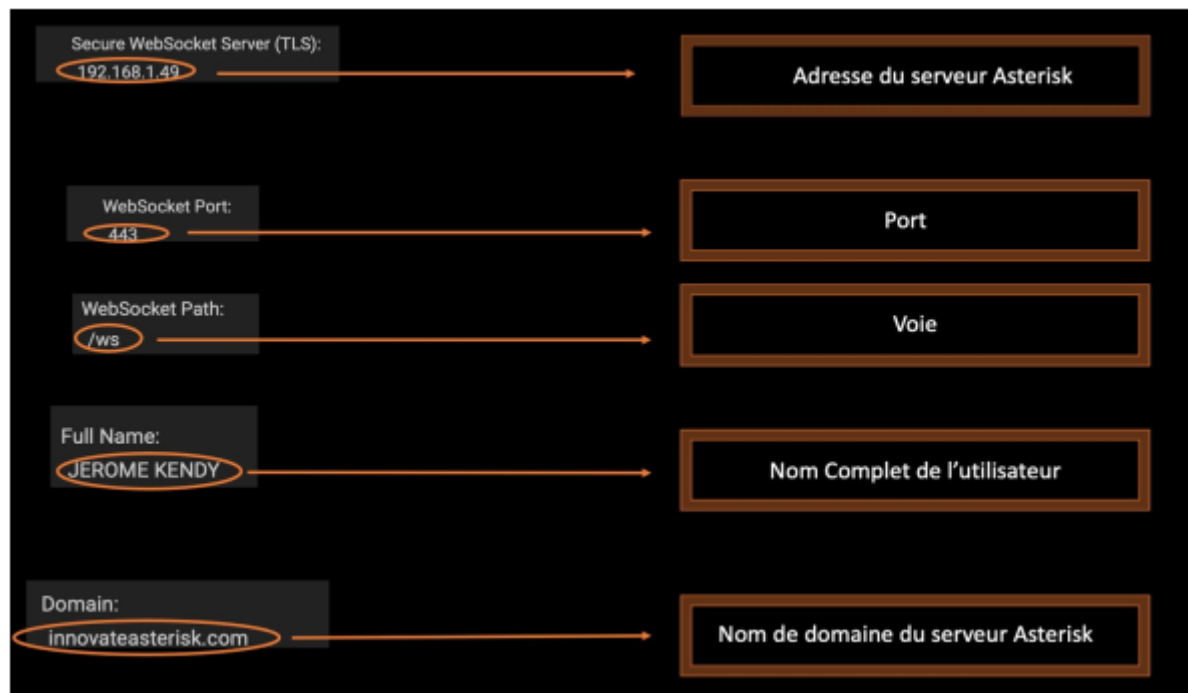
```
;---- Desactiver pjsip.conf ----

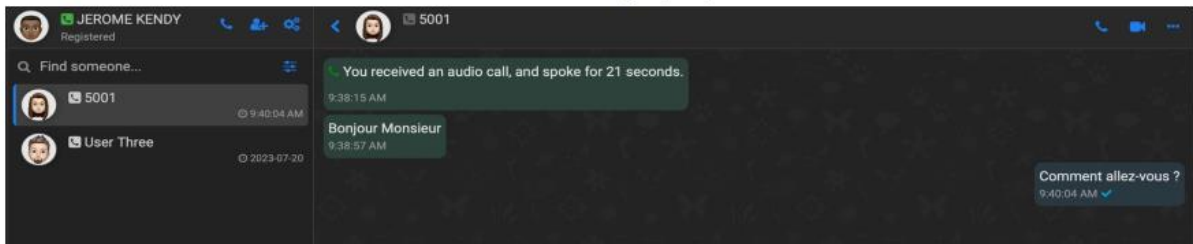
; noload => res_pjsip.so
; noload => res_pjsip_pubsub.so
; noload => res_pjsip_session.so
; noload => chan_pjsip.so
; noload => res_pjsip_exten_state.so
; noload => res_pjsip_log_forwarder.so
```

Après les configurations, vous devez avoir des résultats sur la plateforme Voip. Cette plateforme hébergé sur asterisk possède plusieurs paramètres utiles pour la communication.



Voici les différents paramètres à remplir :





Envoi du message à 5001:

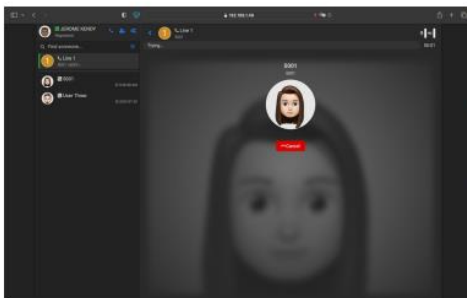
```
Executing [5001@textmessages:1] Gosub("Message/ast_msg_queue", "send-text,s,1,(5001)") in new stack
Executing [s@send-text:1] NoOp("Message/ast_msg_queue", "Sending Text To: 5001") in new stack
Executing [s@send-text:2] Set("Message/ast_msg_queue", "PEER-User1") in new stack
Remote UNIX connection
```

Réception du message envoyer part le User1:

```
Remote UNIX connection disconnected
Executing [s@send-text:3] Set("Message/ast_msg_queue", "FROM-"JEROME KENDY" <100>") in new stack
Executing [s@send-text:4] Set("Message/ast_msg_queue", "CALLERID_NUM-100") in new stack
Executing [s@send-text:5] Set("Message/ast_msg_queue", "FROM_SIP-"JEROME KENDY" <sip:100@innovateasterisk.com>") in new stack
Executing [s@send-text:6] MessageSend("Message/ast_msg_queue", "pjsip:5001,"JEROME KENDY" <sip:100@innovateasterisk.com>") in new stack
```

Message reçue sur le smartphone

---->

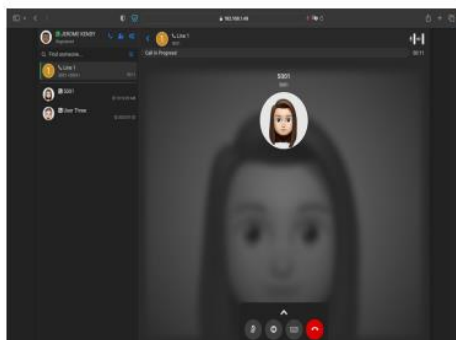


----->

```
Executing [s@dialed-extension:4] Hangup("PJSIP/User1-00000000", "") in new stack
Spawn extension (dial-extension, s, 4) exited non-zero on 'PJSIP/User1-00000000'
```

----->

```
Executing [5001@from-extensions:1] Gosub("PJSIP/User1-00000000", "dial-extension,s,1,(5001)") in new stack
Executing [s@dialed-extension:1] NoOp("PJSIP/User1-00000000", "Calling: 5001") in new stack
Executing [s@dialed-extension:2] Set("PJSIP/User1-00000000", "JITTERBUFFER(adaptive)-default") in new stack
Executing [s@dialed-extension:3] Dial("PJSIP/User1-00000000", "PJSIP/5001,30") in new stack
```



----->

```
Called PJSIP/5001
PJSIP/5001-00000000 is ringing
0x7f70a816c20 -- Strict RTP learning after remote address set to: 192.168.1.33:7242
```

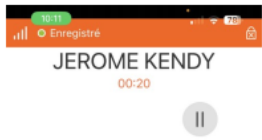
----->

```
Channel PJSIP/User1-00000000 joined 'simple_bridge' basic-bridge <826aecb9-dfic-40d5-8132-83cd623632ba>
0x7f70a816c20 -- Strict RTP learning after ICE completion
0x7f70a816c20 -- Strict RTP learning after remote address set to: 192.168.1.23:51327
0x7f70a816c20 -- Strict RTP switching to RTP target address 192.168.1.23:51327 as source
```

----->

```
0x7f70a816c20 -- Strict RTP learning complete - Locking on source address 192.168.1.23:51327
```

Téléphone Portable via application SIP :



----->

```
PJSIP/5001-00000008 answered PJSIP/User1-00000007
0x7f70a8116c20 -- Strict RTP learning after remote address set to: 92.144.249.205:51327
```



----->

```
Channel PJSIP/5001-00000008 joined 'simple_bridge' basic-bridge <826aecb9-df1c-40d5-8132-83cd623632ba>
```

----->

```
0x7f70a806ba80 -- Strict RTP switching to RTP target address 192.168.1.33:7242 as source
0x7f70a806ba80 -- Strict RTP learning complete - Locking on source address 192.168.1.33:7242
```

