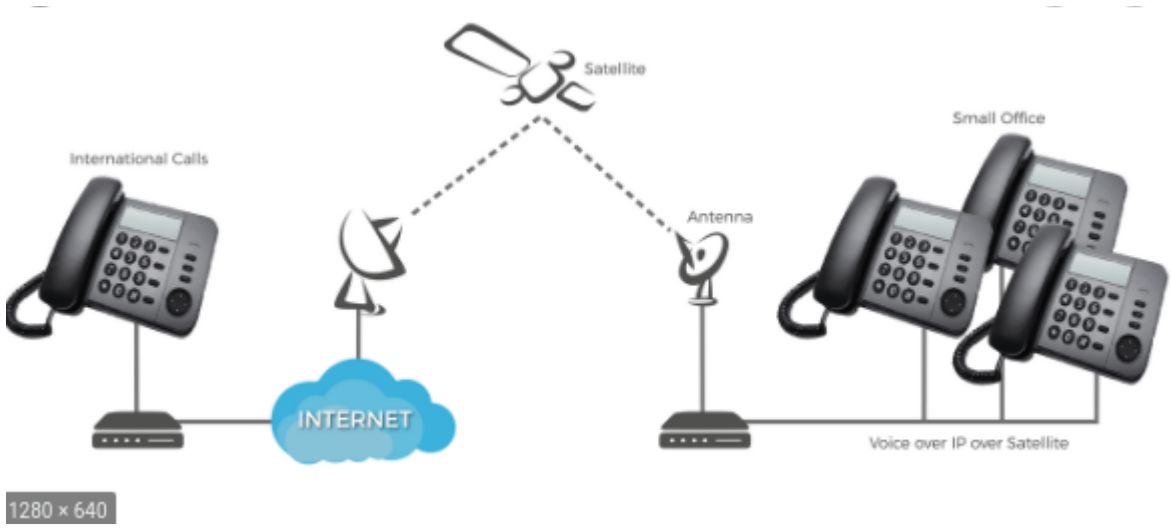
RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)

La VoIP

I - Introduction à la VoIP

La voix sur IP (**VoIP**, **Voice over IP**) est une technologie de communication vocale en pleine émergence. Elle fait partie d'un tournant dans le monde de la communication.
En effet, la convergence du triple play (voix, données et vidéo) fait partie des enjeux principaux des acteurs de la télécommunication. Aujourd'hui au lieu de disposer à la fois d'un réseau informatique et d'un réseau téléphonique commuté (**RTC**), l'entreprise peut donc, grâce à la **VoIP**, tout fusionner sur un même réseau. Comme toute innovation technologique, la VoIP non seulement simplifie l'exploitation des réseaux mais aussi économise de l'argent à ses clients. Les entreprises dépensent énormément en communications téléphoniques, or le prix des communications de la ToIP (Téléphonie sur IP) est dérisoire en comparaison. En particulier, plus les interlocuteurs sont éloignés, plus la différence de prix est intéressante. De plus, la téléphonie sur IP utilise jusqu'à dix fois moins de bande passante que la téléphonie traditionnelle. Il semblerait que les entreprises après avoir émis un certain nombre de doutes sur la qualité de services, soient désormais convaincues de la plus grande maturité technologique des solutions **VoIP** proposées sur le marché. Le scénario vers lequel va s'orienter la téléphonie sur IP dépend beaucoup de l'évolution du réseau lui-même. En effet, si internet garde sa configuration actuelle où elle est essentiellement dimensionnée en fonction d'une qualité de service moyenne pour la transmission des données, cependant cette nouvelle technologie constitue une solution idéale pour le cas d'interconnexion de **PBX** d'entreprises. En effet, les entreprises multi-sites cherchent àéconomiser des dépenses énormes dues aux communications inter-sites. Il serait donc envisageable de profiter de cette nouvelle technologie émergente afin d'assurer cette interconnexion.

I – 1 Avantages de la ToIP



La téléphonie sur IP séduit les entreprises avant tout grâce aux réductions de dépenses qu'elle permet : appels inter-sites gratuits, appels lointains à tarifs très bas, pas de frais de câblage supplémentaires. Mais la téléphonie IP propose encore d'autres avantages…
La téléphonie **IP** permet la transmission des communications vocales via des réseaux au protocole IP (en clair, par Internet dans la plupart des cas) et permet la convergence de tous les postes informatiques d'une entreprise en un seul réseau unifié.Bien.

Mais quelles sont les Avantages concrets pour l'entreprise ?

I – 2 Les principaux atouts de la téléphonie sur IP

- **Coûts d'installation**

Économie sur les frais de câblage si vous possédez déjà un réseau Internet d'entreprise. La téléphonie IP utilise la même connectique. En d'autres termes, pas d'achat de câbles et de frais de pose à assumer.

➢ Coûts de communication

La téléphonie IP demande peu de bande passante. Les entreprises peuvent donc constituer un réseau privé pour relier plusieurs sites distants entre eux. Les communications téléphoniques entre cesderniers sont alors totalement gratuites. La téléphonie IP est donc extrêmement avantageuse et parfaitement adaptée pour les entreprises possédant des implantations géographiques multiples. Pour les appels en dehors de l'entreprise, les coûts sont également bien plus réduits qu'avec la téléphonique traditionnelle. Cela est d'autant plus remarquable pour les appels longue distance, y compris à l'international. Et la distance n'est plus un critère d'augmentation du prix de la communication. A titre d'exemple, un centre d'appels aura les mêmes coûts de communication que ses clients soient tout proches ou disséminés dans le monde entier. Autre facteur de réduction des prix : la concurrence. Les sociétés spécialisées sont nombreuses sur le marché de la téléphonie IP, ce qui créé une pression à la baisse sur les prix. La téléphonie IP fonctionne en associant un utilisateur à un poste et non plus à une ligne. **Qu'est-ce que cela change** ? Un utilisateur peut désormais garder le même numéro de téléphone s'il est en déplacement ou s'il change de bureau.

➢ Simplicité de gestion

La téléphonie traditionnelle obligeait les équipes techniques et informatiques à composer avec des lignes séparées : les données informatiques d'un coté, données **« voix »** de l'autre. Grâce à la technologie **Voix sur IP**, cette distinction disparaît puisqu'il s'agit d'unique et même réseau.
Les opérations de maintenance ou d'évolution sont donc simplifiées, ce qui constitue une source de gains de productivité pour l'entreprise.

➢ Évolutivité

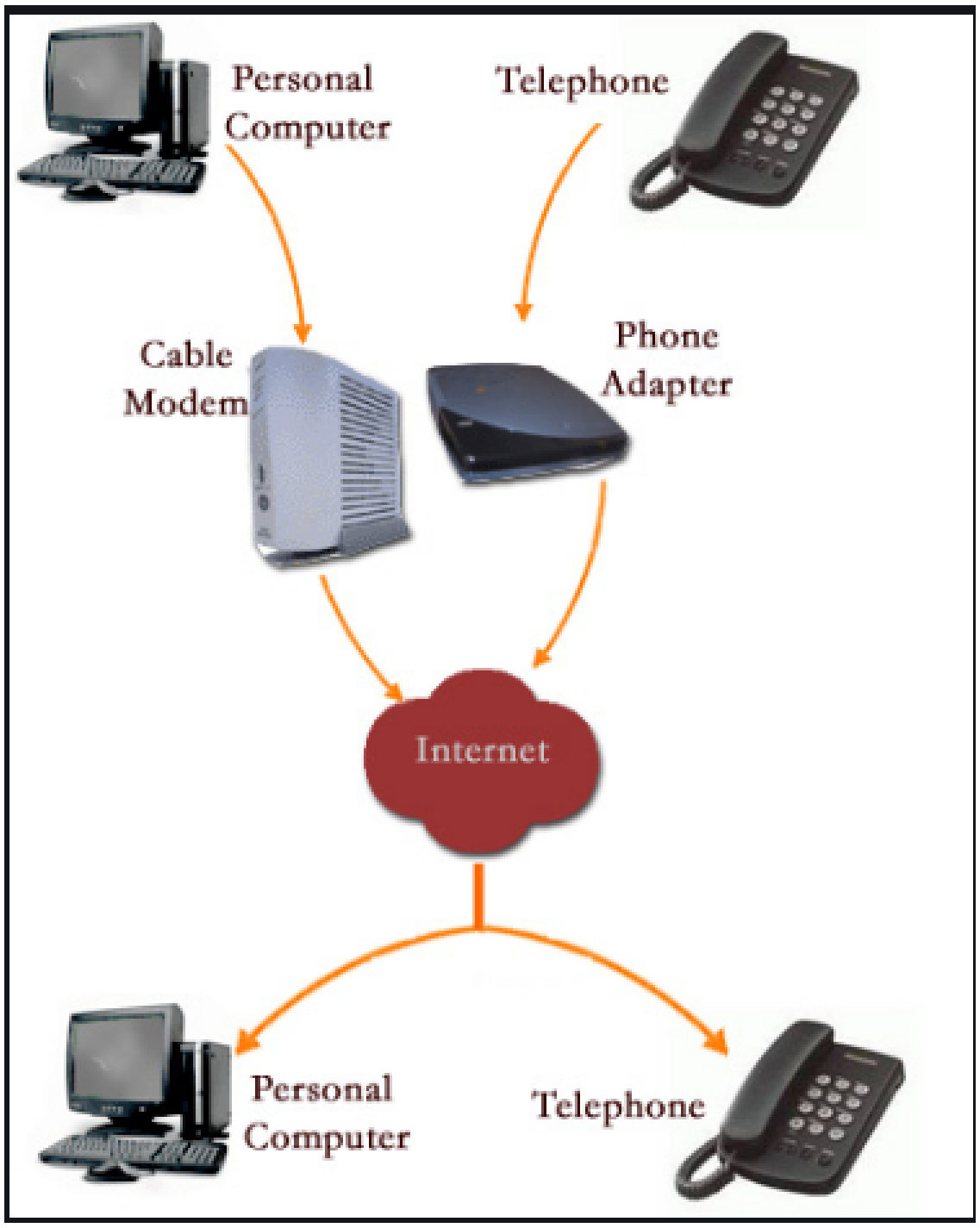
Dans une architecture VoIP, il est très facile d'ajouter des postes supplémentaires ou d'agrandir le réseau, ce qui n'était pas le cas avec la téléphonie non-IP.

II. Asterisk

II – 1. Présentation d’Asterisk

Asterisk est une solution de téléphonie sur IP, Open Source. Il s’agit donc d’une solution gratuite, avec une communauté très active. Il a été développé par la société **Digium**. Asterisk est disponible sous Linux et **Windows**, même si la version **Linux** utilisée est de loin la moins rependue. Il fait office d’**IPBX**, mais il est aussi capable de s’interfacer avec un réseau de téléphonie analogique, à l’aide de cartes additionnelles. Asterisk peut donc faire office d’IPBX et de **PABX**.
Un serveur **Asterisk** permet donc de faire de la VoIP ainsi que de la téléphonie analogique.En plus d’être gratuit, Asterisk est donc aussi très complet et supporte de très nombreux codecs audio, dont le **G.711** (**u-Law** et **a-Law**) et le **G.729**.

II - 2 Architecture généraliste d'Astérisk



II - 3 Fonctionnalités d'Asterisk

Voici une liste non exhaustive des fonctionnalités d'Asterisk

- **Conférence**
- **Musique d'attente**
- **File d'attente**
- **Groupement d'appel**
- **Messagerie vocale**
- **Transfert d'appel**
- **Transcodage entre différents codecs**
- **Notification mail en cas de message (avec message audio en PJ)**
- **Synthèse vocale**
- **Chiffrement des appels**
- **Ne pas déranger**
- **Parking d'appel**
- **Etc…**

II - 4 Quelques fichiers de configuration

- Messagerie vocale **voicemail.conf**
- Parking d'appel **features.conf**
- Transfert d'appels **features.conf**
- Conférence **meetme.conf**
- Centre d'appels **queues.conf** ; **agents.conf**
- Musique d'attente **musiconhold.conf**
- Informations sur les différents répertoires **asterisk.conf**

Installation d'Asterisk

Asterisk est une solution de téléphonie sur IP open source. Développé par la société **Digium**, Il s'agit donc d'une solution gratuite avec une communauté très active. **Asterisk** est disponible sous **Windows** et **Linux**, même si la version Linux utilisée est de loin la moins rependue. Il fait office **d'IPBX**, mais il est aussi capable de s'interfacer avec un réseau de téléphonie analogique, à l'aide de cartes additionnelles.
Un serveur **Asterisk** peut donc faire de la **VoIP** ainsi que de la téléphonie analogique.

Le but de ce TP est de savoir installer convenablement **Asterisk** afin de bénéficier de quasiment toutes les fonctionnalités que nous offre ce logiciel. Note : Ceci est une façon d'installer Asterisk parmi tant d'autres !

infrastructure : Ubuntu 16

On y va…

RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)

Installation des prérequis

apt install libxml2-dev libxml2-utils libcurl4-openssl-dev libeditline-dev libsqlite3-dev curl autoconf gcc g++ build-essential linux-headers-$(uname -r) libssl-dev openssl libc-dev ncurses-dev -y

Téléchargement depuis la source

 # wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz

Après téléchargement, il atterrira dans **~/Téléchargements** , on le désarchive dans un dossier de choix.Personnellement je préfère **/usr/local/src**

```
root@berenger:~/Téléchargements# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
--2021-02-28 15:43:59--  http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
Résolution de downloads.asterisk.org (downloads.asterisk.org)… 76.164.171.238, 2001:470:e0d4::ee
Connexion à downloads.asterisk.org (downloads.asterisk.org)|76.164.171.238|:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 27809393 (27M) [application/x-gzip]
Enregistre : «asterisk-16-current.tar.gz»

asterisk-16-current.tar.gz                0%[                                                                                ] 200,67K  3,13KB/s    tps 2h 42m16
asterisk-16-current.tar.gz              100%[===========================================================================================>]  26,52M  11,0KB/s    ds 31m 11s

2021-02-28 16:15:12 (14,5 KB/s) - «asterisk-16-current.tar.gz» enregistré [27809393/27809393]

root@berenger:~/Téléchargements# ls
asterisk-16-current.tar.gz
root@berenger:~/Téléchargements#
```


```
root@berenger:~/Téléchargements# wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
--2021-02-28 15:43:59--  http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-16-current.tar.gz
Résolution de downloads.asterisk.org (downloads.asterisk.org)… 76.164.171.238, 2001:470:e0d4::ee
Connexion à downloads.asterisk.org (downloads.asterisk.org)|76.164.171.238|:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 27809393 (27M) [application/x-gzip]
Enregistre : «asterisk-16-current.tar.gz»

asterisk-16-current.tar.gz                0%[                                                                                ] 200,67K  3,13KB/s    tps 2h 42m16
asterisk-16-current.tar.gz              100%[===========================================================================================>]  26,52M  11,0KB/s    ds 31m 11s

2021-02-28 16:15:12 (14,5 KB/s) - «asterisk-16-current.tar.gz» enregistré [27809393/27809393]

root@berenger:~/Téléchargements# ls
asterisk-16-current.tar.gz
root@berenger:~/Téléchargements# tar xzf asterisk-16-current.tar.gz -C /usr/local/src/
root@berenger:~/Téléchargements#
```


 # cd /usr/local/src/asterisk-16.16.0

 # contrib/scripts/install_prereq install

Qui doit finir par un truc de ce genre :

```
État actuel : 381 (-40) upgradable.
##################################################
## install completed successfully
##################################################
root@berenger:/usr/local/src/asterisk-16.16.1#
```

./bootstrap.sh

 # ./configure --with-jansson-bundled
ou

./**configure**

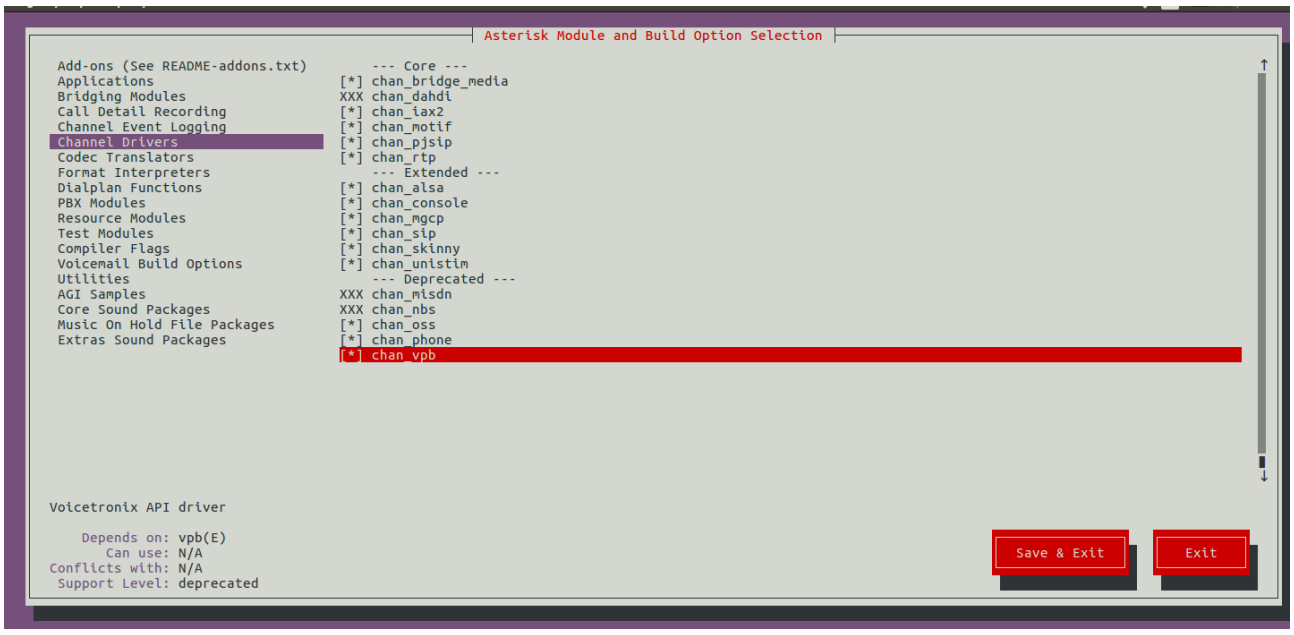
Qui doit finir par un truc de ce genre :

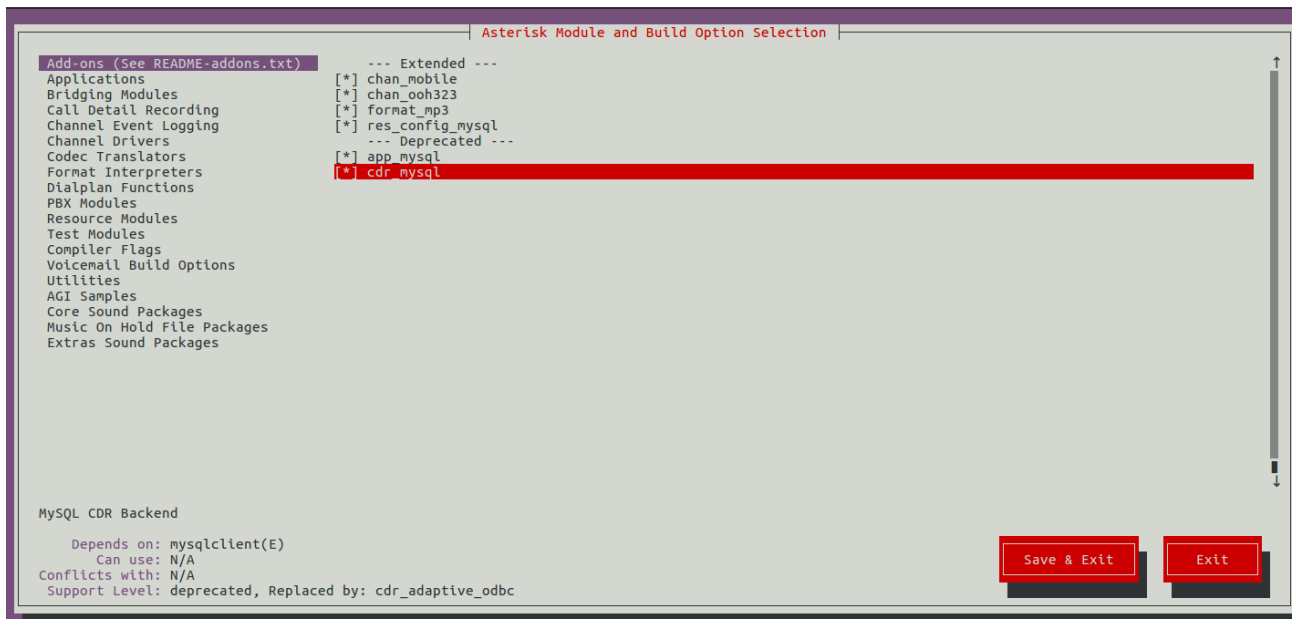
```
configure: Menuselect build configuration successfully completed

              .$$$$$$$$$$$$$$$=..
           .$7$7..          .7$$7:.
         .$$:.                 ,$7.7
       .$7.     7$$$$           .$$77
     ..$$.      $$$$$            .$$$7
    ..7$   .?.  $$$$$   .?.       7$$$.
   $.$.   .$$$7. $$$$7 .7$$$.      .$$$.
 .777.   .$$$$$$77$$$77$$$$$7.      $$$,
 $$$~      .7$$$$$$$$$$$$$7.       .$$$.
.$$7         .7$$$$$$$7:           ?$$$.
$$$         ?7$$$$$$$$$I          .$$$7
$$$       .7$$$$$$$$$$$$$$       :$$$.
$$$       $$$$$$7$$$$$$$$$$    .$$$.
$$$        $$$   7$$$7  .$$$    .$$$.
$$$$             $$$$7         .$$$.
7$$$7            7$$$$       7$$$
 $$$$$                       $$$
  $$$$7.                      $$  (TM)
   $$$$$$.           .7$$$$$  $$
     $$$$$$$$$$$$7$$$$$$$$.$$$$$$
       $$$$$$$$$$$$$$$.

configure: Package configured for:
configure: OS type  : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :
root@berenger:/usr/local/src/asterisk-16.16.1#
```

 # make menuselect

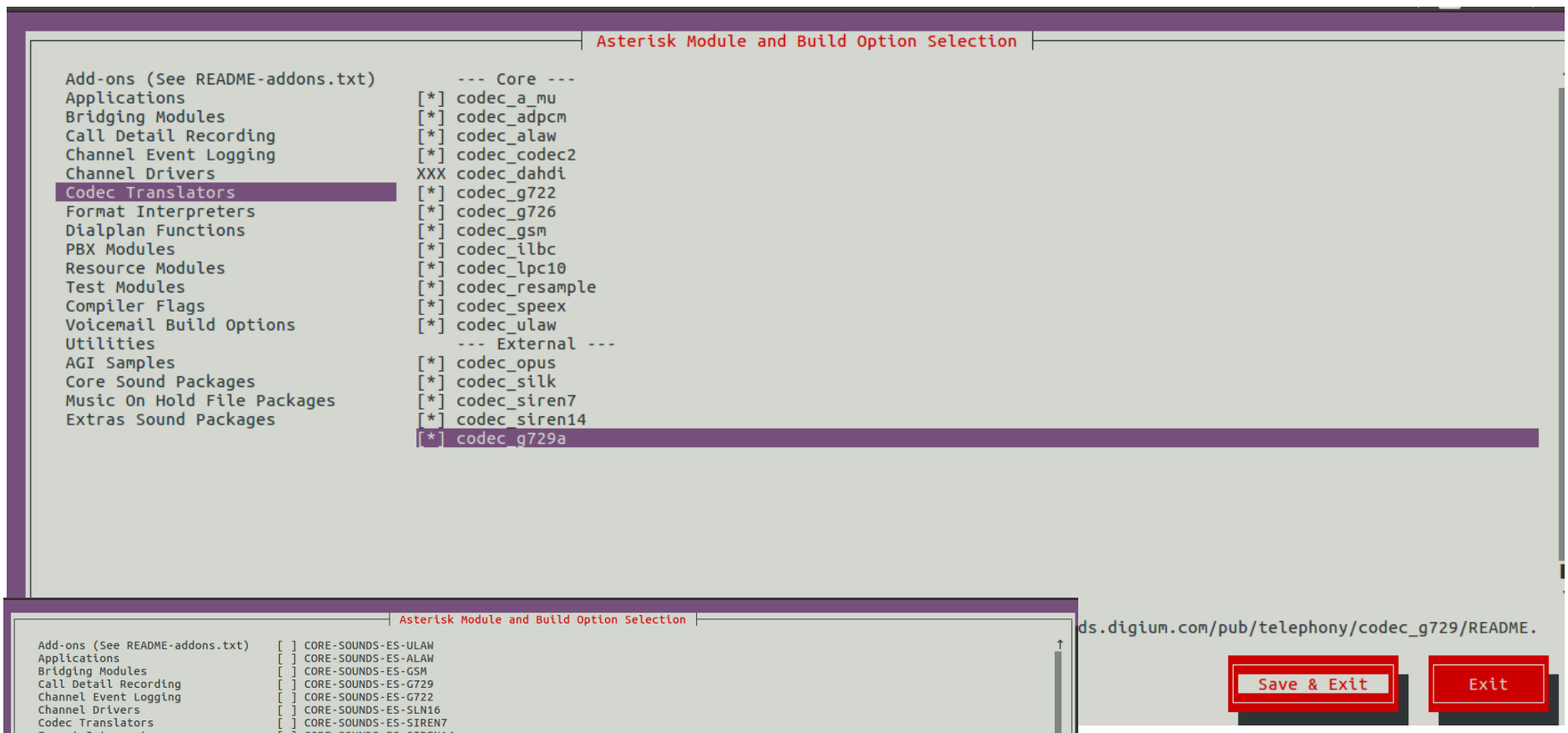
Pour activer certains modules :

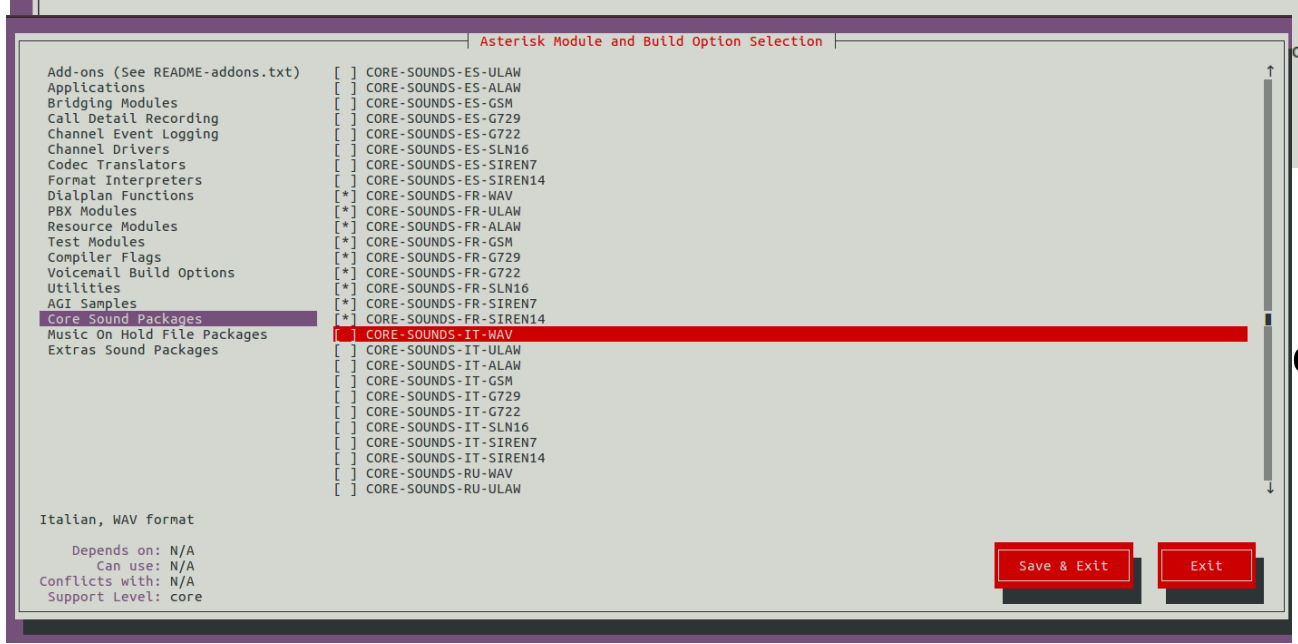
RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)





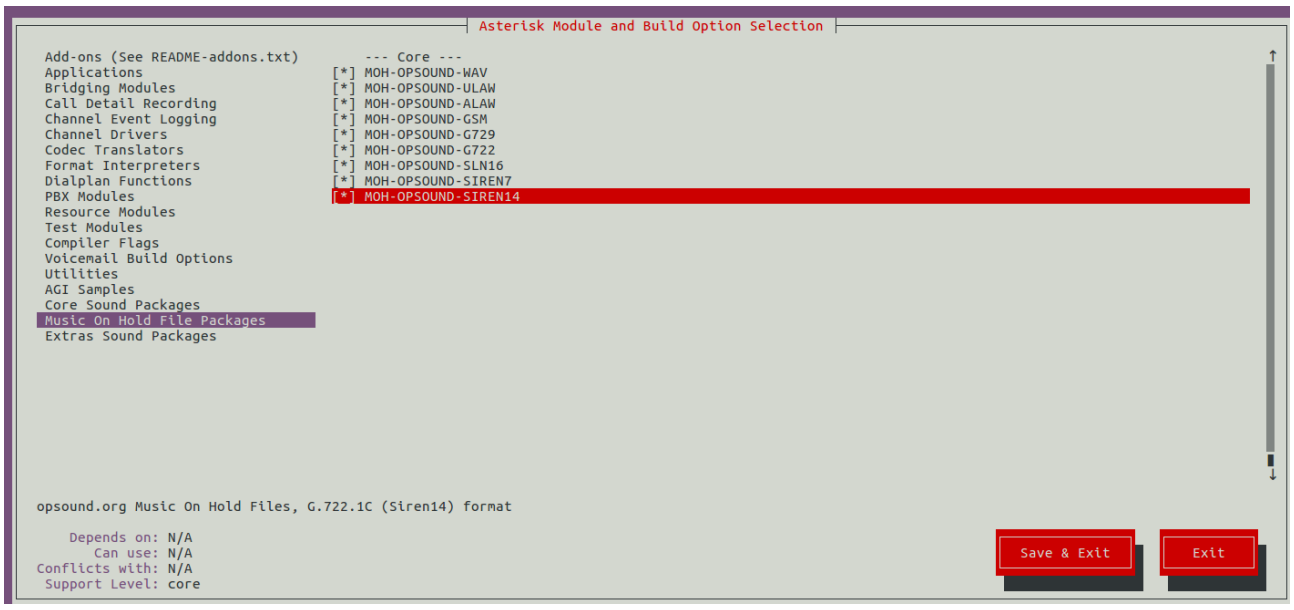
on active les codec audio et **vidéos** :





on active les sons en français

on active les musiques de la salle d’attente :



make

```
Building Documentation For: third-party channels pbx apps codecs formats cdr cel bridges funcs tests main res addons
 +--------- Asterisk Build Complete ---------+
 + Asterisk has successfully been built, and +
 + can be installed by running:              +
 +                                           +
 +                make install               +
 +-------------------------------------------+
root@berenger:/usr/local/src/asterisk-16.16.1#
```

make install

```
root@berenger:/etc/asterisk# ls -1
acl.conf
adsi.conf
agents.conf
alarmreceiver.conf
alsa.conf
amd.conf
app_mysql.conf
app_skel.conf
ari.conf
ast_debug_tools.conf
asterisk.adsi
asterisk.conf
calendar.conf
ccss.conf
cdr_adaptive_odbc.conf
cdr_beanstalkd.conf
cdr.conf
cdr_custom.conf
cdr_manager.conf
cdr_mysql.conf
cdr_odbc.conf
cdr_pgsql.conf
cdr_sqlite3_custom.conf
cdr_syslog.conf
cdr_tds.conf
cel_beanstalkd.conf
cel.conf
cel_custom.conf
cel_odbc.conf
cel_pgsql.conf
cel_sqlite3_custom.conf
cel_tds.conf
chan_dahdi.conf
chan_mobile.conf
cli_aliases.conf
cli.conf
cli_permissions.conf
codecs.conf
confbridge.conf
```

make samples

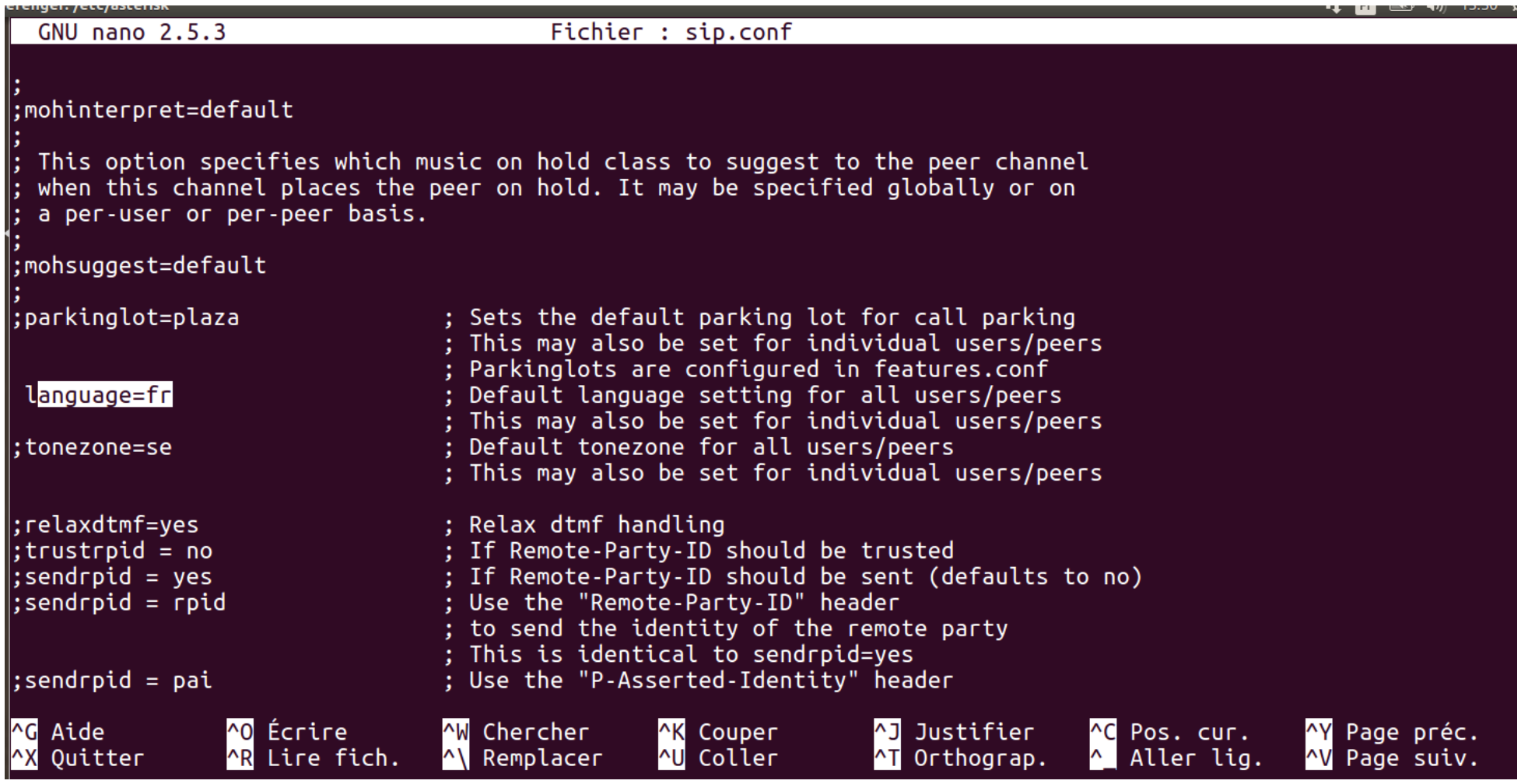
On fait un saut dans le dossier d'Asterisk pour voir s'il est réellement installé

IV – Mise en place de services

Asterisk dispose d’un nombre incalculable de services et nous essayons de valoriser quelques uns :

1- Changement de la langue anglaise par celle française

Rendez-vous dans le fichier sip.conf et chercher la ligne
commençant par **;language=en** , on dé commente la ligne puis on remplace en par fr comme suit :



Nous venons d’activer la langue française !

2- Appels audios

Fichiers concernés

- **sip.conf**
- **extensions.conf**

Dans sip.conf , la création de comptes se fait comme suit :
On fait de même pour créer tout le reste des comptes :

```
[1000]
username = 1000
type = friend
host = dynamic
context = Telephonie
callerid = "Berenger"<1000>
mailbox = 1000@Telephonie
secret = bangui
```

Rendez-vous dans extensions.conf pour définir le plan de numérotation .

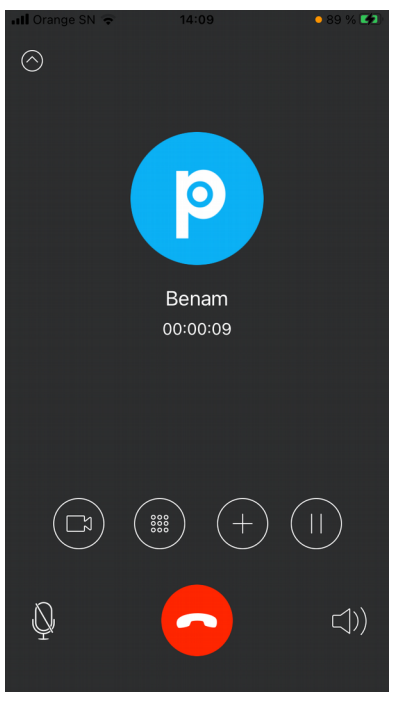
```
[Telephonie]
exten => _1XXX,1,Dial(SIP/${EXTEN},10,tr);le phone sonne pendant 10 secondes
```

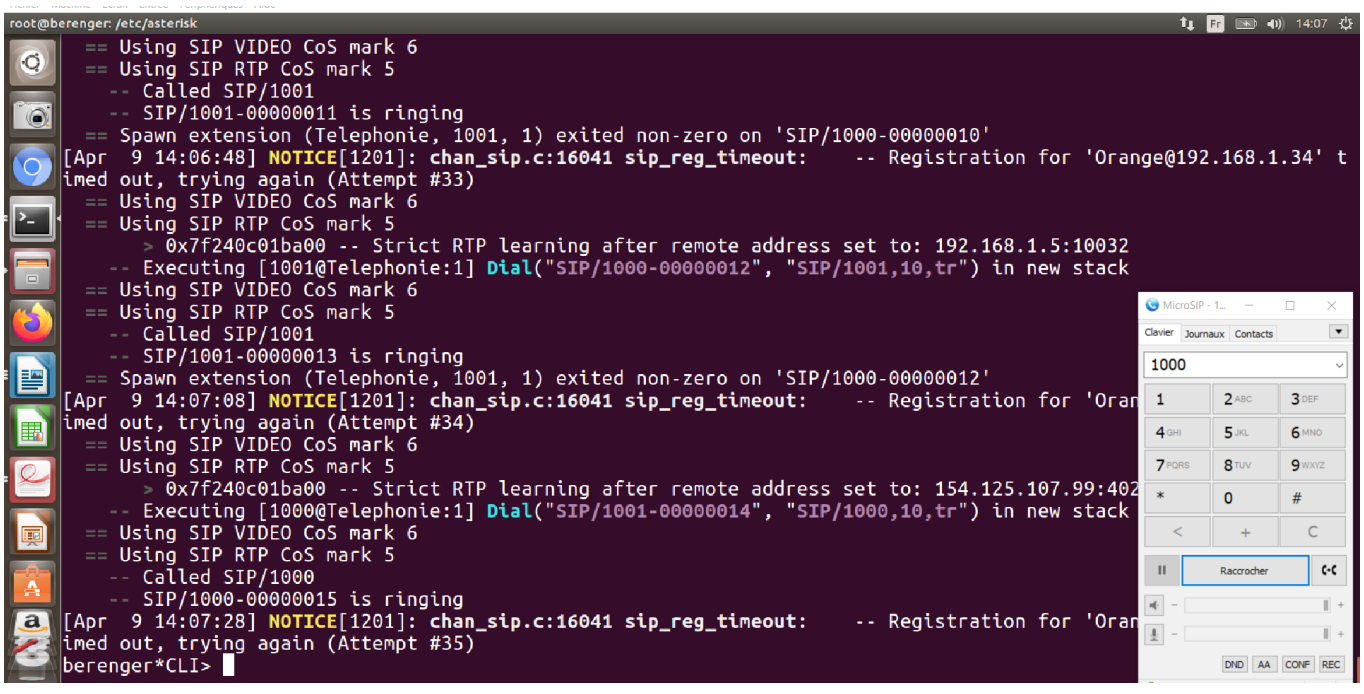
On sauvegarde la config puis on lance Asterisk en mode CLI
comme suit :

```
root@berenger:/etc/asterisk# asterisk -rvvvvvvvvvv
Asterisk 16.16.1, Copyright (C) 1999 - 2018, Digium, Inc. 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 16.16.1 currently running on berenger (pid = 984)
[Apr  9 13:53:23] NOTICE[1201]: chan_sip.c:16041 sip_reg_timeout:     -- Registration for 'Orange@192.168.1.3
4' timed out, trying again (Attempt #190)
berenger*CLI>
```

Avec ce mode, nous saurons ce qui se passe en temps réel .

Passons au Test d’appel :





Ici le user 1000 reçoit l’appel de 1001

2. Messagerie vocale

Fichiers concernés

- sip.conf
- voicemail.conf
- extensions.conf

➢ sip.conf dans lequel on met cette ligne pour chaque utilisateur :

```
mailbox = 1000@Telephonie
```

Idem pour user 1001

```
mailbox = 1000@Telephonie
```

Cette ligne doit être présente au niveau de tous les comptes activés : le seul changement sera au niveau du numéro avant @ . 1000 sera ce qu'on mettra dans ce compte, et ainsi de suite .

➢ Voicemail.conf dans lequel on met ceci :

```
[BV]
1000 =>1234
1001 =1234
```

[BV] est notre contexte de toujours, depuis la création des comptes, (nous l'utiliserons tout au long de ce **TP**)
1000 => mot_de_passe,nom_d'utilisateur # ces deux options sont facultatives, (dans notre cas elles n'existent même pas)

C'est la même explication pour la deuxième ligne (pour le compte 1001)

➢ **extensions.conf** dans lequel on met ceci :

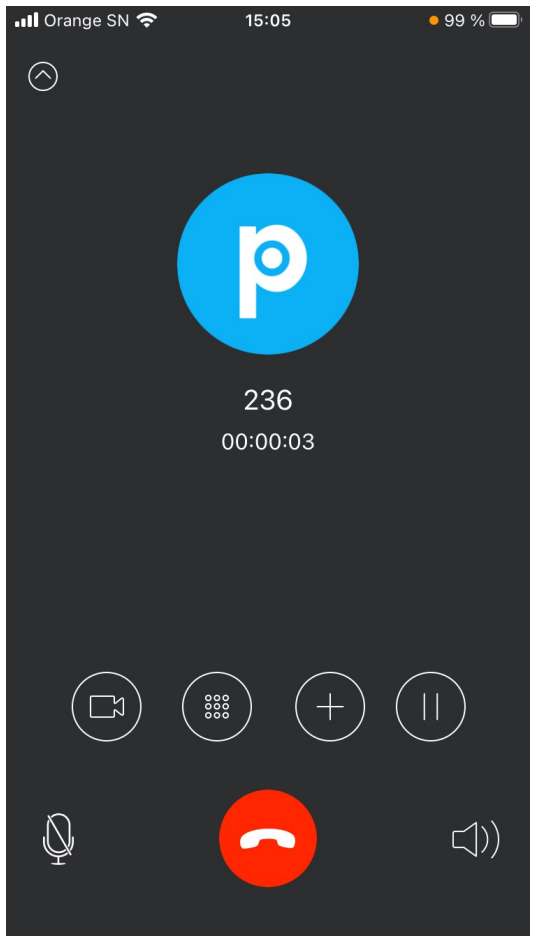
```
[Telephonie]
exten => _1XXX,1,Dial(SIP/${EXTEN},10,tr);le phone sonne pendant 10 secondes
exten => _1XXX,2,Voicemail(${EXTEN}@BV)
exten => _236,1,VoicemailMain(${CALLERID(num)}@Telephonie)
```

Explication

◆ Ici, 236 est le numéro à composé pour écouter ses messages vocaux

◆ **VoiceMailMain** est la fonction qui permet d'activer la messagerie vocale .

Voici la preuve que le numéro **236** existe bel et bien



Quand nous composons le numéro de la boîte vocale, côté serveur nous aurons ceci :

```
sterisk/voicemail/BV/1000/INBOX/msg0001 failed
    -- <SIP/1000-00000028> Playing 'vm-prev.ulaw' (language 'fr')
  == Spawn extension (Telephonie, 236, 1) exited non-zero on 'SIP/1000-00000028'
  == Using SIP VIDEO CoS mark 6
  == Using SIP RTP CoS mark 5
       > 0x7f240c062db0 -- Strict RTP learning after remote address set to: 192.168.1.5:10024
    -- Executing [236@Telephonie:1] VoiceMailMain("SIP/1000-00000029", "1000@BV") in new stack
    -- <SIP/1000-00000029> Playing 'vm-password.ulaw' (language 'fr')
       > 0x7f240c062db0 -- Strict RTP switching to RTP target address 192.168.1.5:10024 as source
```

Transfert d'appels

Son principe est simple. Comme son nom l'indique, il permet de transférer un appel encours vers un autre numéro.

Fichiers concernés

- **features.conf**
- **extensions.conf**

Dans features.conf

```
[featuremap]
;blindxfer => #1                ; Blind transfer  (default is #) -- Make sure to set the T and/or t option $
;disconnect => *0               ; Disconnect  (default is *) -- Make sure to set the H and/or h option in t$
;automon => *1                  ; One Touch Record a.k.a. Touch Monitor -- Make sure to set the W and/or w $
;atxfer => *2                   ; Attended transfer  -- Make sure to set the T and/or t option in the Dial($
```

Nous localisons le contexte featuremap, et nous apportons les modifications suivantes :

```
[featuremap]
blindxfer => ##                ; Blind transfer  (default is #) -- Make sure to set the T and/or t option i$
;disconnect => *0               ; Disconnect  (default is *) -- Make sure to set the H and/or h option in t$
;automon => *1                  ; One Touch Record a.k.a. Touch Monitor -- Make sure to set the W and/or w $
atxfer => **                   ; Attended transfer  -- Make sure to set the T and/or t option in the Dial()$
;parkcall => #72                ; Park call (one step parking)  -- Make sure to set the K and/or k option i$
;automixmon => *3               ; One Touch Record a.k.a. Touch MixMonitor -- Make sure to set the X and/or$
```

- **blindxfer** => ## ; pour les transferts d'appels aveugles
- **atxfer** => ** ; pour les transferts d'appels guidés

Dans extensions.conf

```
[Telephonie]
exten => _1XXX,1,Dial(SIP/${EXTEN},10,tT);le phone sonne pendant 10 secondes
exten => _1XXX,2,Voicemail(${EXTEN}@BV)
exten => _236,1,VoicemailMain(${CALLERID(num)}@BV)
```

Nous avons ajouté les options **tT** pour activer le transfert .
Après toute cette configuration, nous devrions nous demander comment pourrait-on transférer un appel ….

Explication

Au cours d'un appel, si un user compose le ## , l'appel encours sera transféré à un numéro aléatoire. C'est pourquoi nous l'appelons transfert aveugle. Par contre, s'il compose ** , le système lui demandera de saisir le numéro auquel transférer l'appel. D'où son nom transfert guidé.

La fonction record

Cette fonction nous permet nous administrateur du réseau téléphonique de faire des préenregistrements.

Fichiers concernés

- **extensions.conf**
- **/var/lib/asterisk/sounds/fr/…**

Dans **extensions.conf ,** nous ferons ceci :

```
;record
exten => 333,1,Answer()
exten => 333,2,Wait()
exten => 333,3,Record(/var/lib/asterisk/sounds/fr/Benam.gsm)
```

Explication

- **333** # c'est le numéro à composer pour bénéficier à ce service
- **333**,**1**,**Answer()**

dans un premier temps, le serveur décroche l'appel ;

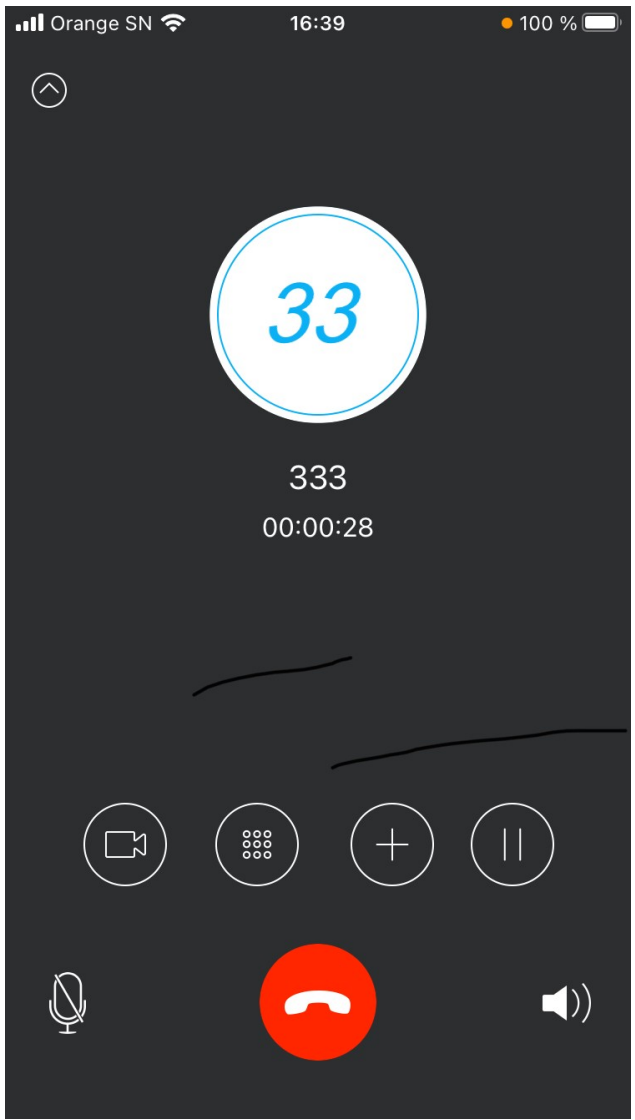
- **333,2,wait()** # dans un second temps, le serveur attends un

instant pour nous permettre d'enregistrer notre voix ;

- **333,3,Record**

Après avoir enregistré notre voix, cette dernière sera stockée dans un fichier appelé Berenger.gsm qui sera dans le répertoire **/var/lib/asterisk/sounds/fr/**

Place au test :



Et nous aurons ceci :

```
root@berenger:/var/lib/asterisk/sounds# ls fr/ | grep ^Berenger
Berenger.gsm
root@berenger:/var/lib/asterisk/sounds# ls fr/ | grep ^Berenger
```

Nous constatons que le fichier Benam.gsm a bel et bien été créé .C'est bien beau de pouvoir enregistrer notre voix, mais ça ne sert à rien si nous pouvons pas la réécouter ; c'est ce qui d'ailleurs sera la prochaine fonction .

- 6. **La fonction Playback**
  Cette fonction permet de lire un préenregistrement

Fichier concerné

◆ **extensions.conf**

Ici **extensions.conf** est le seul fichier qui intervient, et nous ferons ceci :

```
;Playback
exten => 334,1,Playback(berenger)
```

Explication

Cette ligne suffit largement pour mettre en place cette option

◆ **334** # numéro à composer pour y accéder

◆ **Playback** # Fonction

◆ (**benam**)

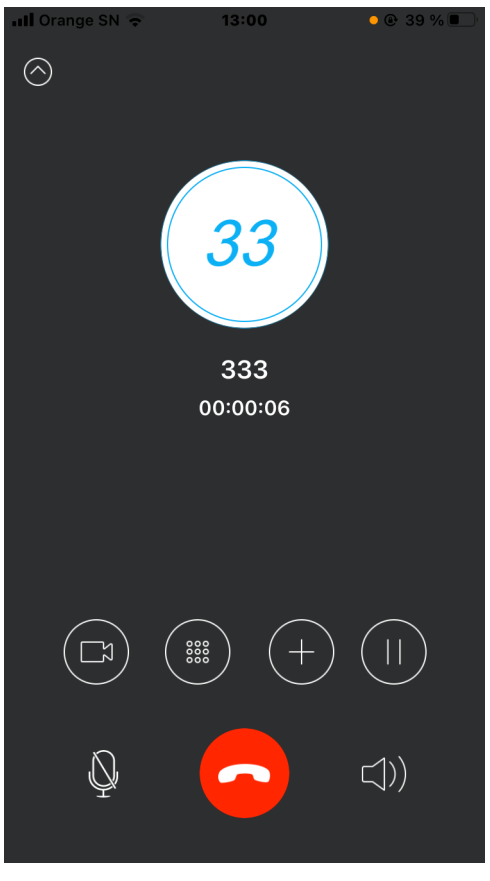
c'est le fichier contenant la voix qui est stocké dans répertoire **/var/lib/asterisk/sounds/fr/**

Place au test

Côté serveur :

```
  == Using SIP RTP CoS mark 5
       > 0x7fd7c8008950 -- Strict RTP learning after remote address set to: 192.168.1.4:10000
    -- Executing [334@Telephonie:1] Playback("SIP/1000-00000005", "berenger") in new stack
```

Côté client :



7. La Conférence

principe

Cette fonction comme son nom l'indique, permet à travers un numéro à composer et un code à saisir de mettre plusieurs abonnés en conférence .

Fichiers concernés

- **confbridge.conf**
- **extensions.conf**

La configuration se fait comme suit :

Dans **confbridge.conf**

```
[admin_user]
type = user
pin = 1234
marked = yes
music_on_hold_when_empty = yes
announce_user_count = yes

[default_user]
type = user
wait_marked = yes
pin = 1234
music_on_hold_when_empty = yes
announce_user_count = yes

[default_bridge]
type = bridge
max_members = 20
```

Explication de la configuration

[admin_user]

- **type = user** # avant tout, l'administrateur est aussi un utilisateur
- **pin = 1234 #** 1234 est son mot de passe pour ouvrir la session
- **music_on_hold_when_empty = yes** # Une musique d'attente sera exécutée lorsque la salle est vide
- **announce_user_count = yes** # annoncer le nombre de participants dans la salle de conférence

[default_user] # Contexte utilisateurs

- **pin = 1234** # Mot de passe à saisir pour accéder à la conférence

[default_bridge] # Contexte du pont

- **type = bridge** # C'est le lien entre les users pour former une salle
- **max_members = 20** # nombre maximum de participants

Dans extensions.conf

```
;Conference
exten => 400,1,Progress()
exten => 400,2,Wait(1)
exten => 400,3,ConfBridge(1,default_bridge,admin_user)

exten => 401,1,Progress()
exten => 401,2,Wait(1)
exten => 401,3,ConfBridge(1,default_bridge,default_user)
```

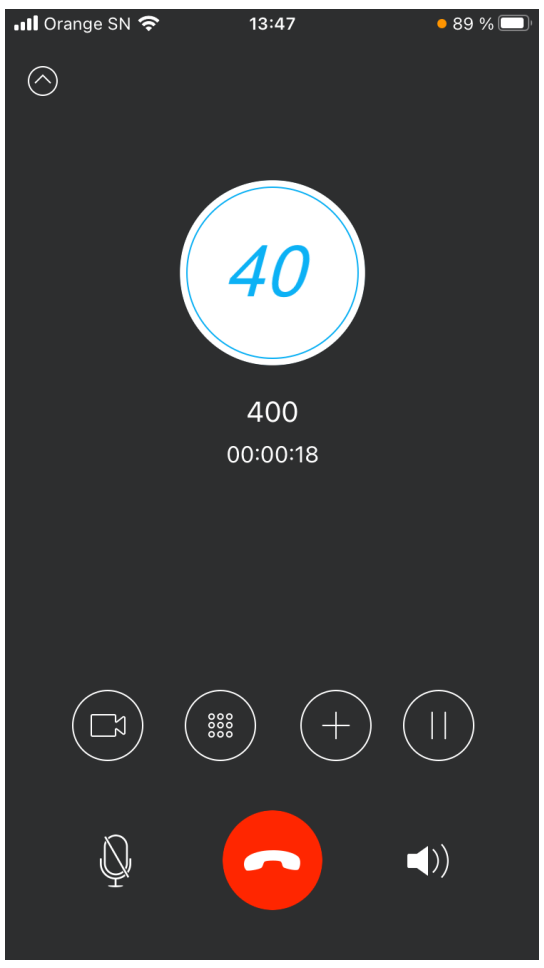
Mécanisme

Ici l'administrateur n'a qu'à composer **400** pour ouvrir la conférence ; suivi de son mot de passe pour s'authentifier. Les **users** quant à eux doivent composer **401** pour y intégrer. Naturellement suivi de leur mot de passe pour s'authentifier.

Coté Serveur :

```
       > 0x7fd7c8011220 -- Strict RTP learning after remote address set to: 192.168.1.6:10016
    -- Executing [400@Telephonie:1] Progress("SIP/1000-0000000a", "") in new stack
    -- Executing [400@Telephonie:2] Wait("SIP/1000-0000000a", "1") in new stack
       > 0x7fd7c8011220 -- Strict RTP switching to RTP target address 192.168.1.6:10016 as source
    -- Executing [400@Telephonie:3] ConfBridge("SIP/1000-0000000a", "1,default_bridge,admin_user") in new stack
    -- <SIP/1000-0000000a> Playing 'conf-getpin.ulaw' (language 'fr')
       > 0x7fd7c8011220 -- Strict RTP learning complete - Locking on source address 192.168.1.6:10016
    -- Channel CBAnn/1-00000001;2 joined 'softmix' base-bridge <0f7788eb-dfbb-4788-94da-bbabec851d38>
    -- <SIP/1000-0000000a> Playing 'conf-onlyperson.ulaw' (language 'fr')
[Apr 17 13:30:08] NOTICE[1543]: chan_sip.c:16041 sip_reg_timeout:     -- Registration for 'Orange@192.168.1.34' t
imed out, trying again (Attempt #2001)
```

son terminal :



8 . Le parking d’appels

Principe

Le Parking Call est la possibilité de mettre en attente une personne sur un numéro de parking, c’est à dire que vous transférez cette personne sur un numéro qui lui sera attribué, dès lors il vous suffit de rappeler ce numéro à partir de n’importe quel poste pour reprendre l’appel.

Les fichiers concernés

- **features.conf**
- **extensions.conf**

Dans features.conf nous mettrons ceci :

RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)

```
[general]
parkext=700
parkpos=701-720
context=parkedcalls
;transferdigittimeout => 3      ; Number of seconds to wait between digits when transferring a call
```

Dans **extensions.conf**

```
[Telephonie]
include => parkedcalls
```

Fonctionnement

Toute fois, un client voulant **parker** l'appel d'un autre, n'a qu'à composer **700##** , le système lui annoncera un numéro compris de **701** à **720** qui servira à récupérer l'appel à partir de n'importe quel poste du réseau. On peut tout de même définir une musique d'attente qui sera exécutée lorsqu'il n'y a personne dans la salle de conférence .

9 . Musique d'attente

Comme son nom l'indique, le principe de la musique d'attente permet à un client d'écouter de la musique lorsque son appel est en attente.

Fichier concerné

- musiconhold.conf ;

9.1 Version wav

Avec cette version, ce sont les musiques du système qui seront exécutées.

Configuration

Au niveau du contexte **default** dans lequel nous mettrons les trois lignes suivantes :

```
[default]
mode = files
directory = /var/lib/asterisk/moh_
random = yes
```

Explication

- **mode=files** # pour indiquer que ce sont des fichiers qui seront exécutés
- **directory=/var/lib/asterisk/moh** # indique le chemin absolu des fichier son à exécuter
- **random=yes** # Utilisation aléatoire des fichiers (la sélection du fichier à exécuter est aléatoire)

9.2 Version mp3

On peut aussi remplacer la version **wav** par des musiques de notre choix, de préférence d'extension mp3 rendez-vous dans le répertoire **/var/lib/asterisk/** pour y créer un répertoire du nom de **mohmp3** dans lequel copier nos fichiers **mp3**

```
root@berenger:/var/lib/asterisk/mohmp3# ls
RCA.mp3
root@berenger:/var/lib/asterisk/mohmp3#
```

Rendez-vous dans **musiconhold.conf** , toujours dans le contexte default pour y faire quelques modifications :

```
[default]
mode = files
directory = /var/lib/asterisk/mohmp3
random = yes
```

Puis qu'il n'y a qu'un seul fichier son dans le dossier, c'est pas la peine d'activer le paramètre **random** .

Note : Cette musique d'attente s'exécutera lorsque nous intégrons une salle de conférence vide ou lorsque notre appel est en attente.

10. Interconnexion de serveurs asterisk

Le but de l'interconnexion de serveurs Asterisk est de permettre à des abonnés d'un serveur (A) de pouvoir communiquer avec des abonnés d'un serveur (B).

Fichiers concernés

- **sip.conf**
- **extensions.conf**

Soit deux serveurs :

Serveur(A)

Dans sip.conf :

```
[serveur(B)]
username = serveur(B)
type = friend
host = 192.168.1.10
qualify = yes
trunk = yes
insecure = port,invite
secret = passer
context = telephonie
```

Explication 1

[serveur(A)] # Contexte qu'on a créé pour l'autre opérateur
host=192.168.1.4 # Son adresse IP
secret=passer # Son mot de passe
context=telephonie # C'est le contexte du serveur B Après cette petite config, rendez-vous dans le **contexte general** ,toujours dans le fichier :

```
[general]
register=>serveur(B):passer@192.168.1.10
context=public                  ; Default context for incoming calls. Defaults to 'd$
;allowguest=no                  ; Allow or reject guest calls (default is yes)
```

Explication 2

nom_serveur(A):son_mot_de_passe@son_adresse_IP
register=>serveur(B):passer@192.168. 1.4 #

Dans extensions.conf

```
exten => _2XXX,1,Dial(SIP/serveur(A)/${EXTEN},20,tT)
```

Note : Les numéros d’appels du serveur (A) commencent par 2 ;

Nous ferons la même chose pour le serveur(B)

```
[serveur(B)]
username = serveur(B)
type = friend
host = 192.168.1.10
qualify = yes
trunk = yes
insecure = port,invite
secret = passer
context = telephonie
```

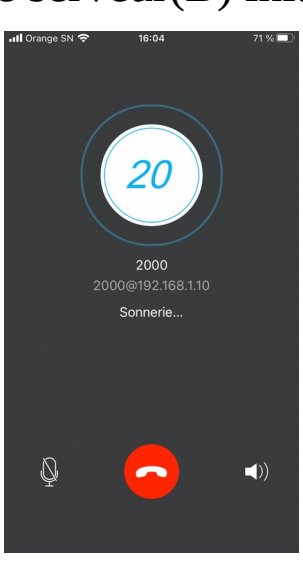
```
[general]
register=>serveur(B):passer@192.168.1.10
context=public                  ; Default context for incoming calls. Defaults to 'd$
;allowguest=no                  ; Allow or reject guest calls (default is yes)
```

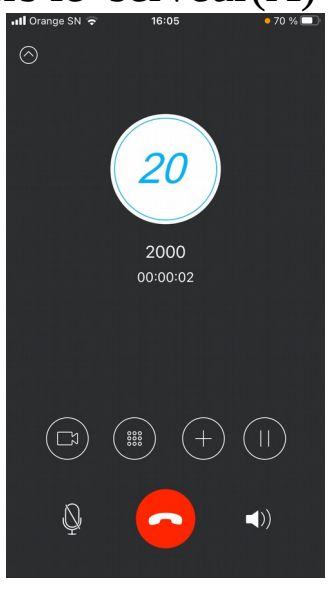
Dans extensions.conf :

```
exten => _1XXX,1,Dial(SIP/serveur(B)/${EXTEN},20,Tt)
```

Place au test

Le serveur(B) initie un appel vers le serveur(A)





coté du serveur(B)

RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)

```
== Using SIP RTP CoS mark 5
     > 0x25c94c0 -- Strict RTP learning after remote address set to: 192.168.1.6:10004
  -- Executing [2000@Telephonie:1] Dial("SIP/1000-00000001", "SIP/serveur(A)/2000,20,tT") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
  -- Called SIP/serveur(A)/2000
  -- SIP/serveur(A)-00000002 is ringing
     > 0x7fc9e4023260 -- Strict RTP learning after remote address set to: 192.168.1.4:11974
  -- SIP/serveur(A)-00000002 answered SIP/1000-00000001
  -- Channel SIP/serveur(A)-00000002 joined 'simple_bridge' basic-bridge <5f3374ec-7a93-4a4b-bd99-62b4929e6941

  -- Channel SIP/1000-00000001 joined 'simple_bridge' basic-bridge <5f3374ec-7a93-4a4b-bd99-62b4929e6941>
     > 0x25c94c0 -- Strict RTP switching to RTP target address 192.168.1.6:10004 as source
     > 0x25c94c0 -- Strict RTP learning complete - Locking on source address 192.168.1.6:10004
     > 0x7fc9e4023260 -- Strict RTP switching to RTP target address 192.168.1.4:11974 as source
  -- Channel SIP/1000-00000001 left 'simple_bridge' basic-bridge <5f3374ec-7a93-4a4b-bd99-62b4929e6941>
== Spawn extension (Telephonie, 2000, 1) exited non-zero on 'SIP/1000-00000001'
  -- Channel SIP/serveur(A)-00000002 left 'simple_bridge' basic-bridge <5f3374ec-7a93-4a4b-bd99-62b4929e6941>
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
     > 0x25deb20 -- Strict RTP learning after remote address set to: 192.168.1.6:10008
  -- Executing [2000@Telephonie:1] Dial("SIP/1000-00000003", "SIP/serveur(A)/2000,20,tT") in new stack
== Using SIP VIDEO CoS mark 6
```

```
  -- Executing [2000@Telephonie:1] Dial("SIP/1000-00000003", "SIP/serveur(A)/2000,20,tT") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
  -- Called SIP/serveur(A)/2000
  -- SIP/serveur(A)-00000004 is ringing
     > 0x7fc9cc00bd40 -- Strict RTP learning after remote address set to: 192.168.1.4:18882
  -- SIP/serveur(A)-00000004 answered SIP/1000-00000003
  -- Channel SIP/serveur(A)-00000004 joined 'simple_bridge' basic-bridge <8d155494-f30d-4d0a-b48a-a4821315128e

  -- Channel SIP/1000-00000003 joined 'simple_bridge' basic-bridge <8d155494-f30d-4d0a-b48a-a4821315128e>
     > 0x25deb20 -- Strict RTP switching to RTP target address 192.168.1.6:10008 as source
  -- Channel SIP/serveur(A)-00000004 left 'simple_bridge' basic-bridge <8d155494-f30d-4d0a-b48a-a4821315128e>
  -- Channel SIP/1000-00000003 left 'simple_bridge' basic-bridge <8d155494-f30d-4d0a-b48a-a4821315128e>
== Spawn extension (Telephonie, 2000, 1) exited non-zero on 'SIP/1000-00000003'
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
     > 0x25deb20 -- Strict RTP learning after remote address set to: 192.168.1.6:10012
  -- Executing [2000@Telephonie:1] Dial("SIP/1000-00000005", "SIP/serveur(A)/2000,20,tT") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
```

Asterisk : Mettre en place un menu intéractif (IVR) avec GoogleTTS

Vous avez déjà du, en appelant un service administratif ou le **SAV** d’un magasin, tomber sur une voix robotisé qui vous demandais de taper sur des touches de votre clavier téléphonique pour joindre un service spécifique?

Vous trouvez ça énervant? Hé bien c’est ce que nous allons mettre en place on y va!!!!!!! !

Qu’est-ce qu’un IVR?

Un **IVR** (**I**nteractive **V**oice **R**esponse) permet de guider les utilisateurs vers le service qu’ils souhaitent joindre, on peut s’en servir aussi pour simplement annoncer les horaires d’ouvertures etc...

La mise en place d’un IVR va se faire en deux étapes, la première il faut que nous apprenons à Asterisk à nous parler, la seconde est la création de l’IVR.

Comment faire parler Asterisk?

Pour faire parler Asterisk, nous allons installer un moteur de synthèse vocale (Text To Speech) et nous allons mettre en place celui de Google appelé **GoogleTTS**.

Installation de GoogleTTS

Tout d’abord GoogleTTS à besoins de quelques dépendances pour fonctionner:
#apt-get install perl libwww-perl sox mpg123

```
root@berenger:~# apt-get install perl libwww-perl sox mpg123
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
libwww-perl est déjà la version la plus récente (6.15-1).
perl est déjà la version la plus récente (5.22.1-9ubuntu0.9).
mpg123 est déjà la version la plus récente (1.22.4-1ubuntu0.1).
sox est déjà la version la plus récente (14.4.1-5+deb8u4ubuntu0.1).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  linux-headers-4.15.0-139 linux-headers-4.15.0-139-generic linux-image-4.15.0-139-generic
  linux-modules-4.15.0-139-generic linux-modules-extra-4.15.0-139-generic snapd-login-service
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 2 non mis à jour.
root@berenger:~#
```

Puis nous allons aller dans le dossier **/var/lib/asterisk/agi-bin/** et y mettre le fichier **googletts.agi** qui est le script pour mettre en place la synthèse vocale.

```
cd /var/lib/asterisk/agi-bin
wget https://raw.github.com/zaf/asterisk-googletts/master/googletts.agi
chmod +x googletts.agi
```

```
root@berenger:~# cd /var/lib/asterisk/agi-bin
root@berenger:/var/lib/asterisk/agi-bin# wget https://raw.github.com/zaf/asterisk-googletts/master/googletts.agi
--2021-04-21 21:45:11--  https://raw.github.com/zaf/asterisk-googletts/master/googletts.agi
Résolution de raw.github.com (raw.github.com)… 185.199.109.133, 185.199.110.133, 185.199.108.133, ...
Connexion à raw.github.com (raw.github.com)|185.199.109.133|:443… connecté.
requête HTTP transmise, en attente de la réponse… 301 Moved Permanently
Emplacement : https://raw.githubusercontent.com/zaf/asterisk-googletts/master/googletts.agi [suivant]
--2021-04-21 21:45:13--  https://raw.githubusercontent.com/zaf/asterisk-googletts/master/googletts.agi
Résolution de raw.githubusercontent.com (raw.githubusercontent.com)… 185.199.108.133, 185.199.109.133, 185.199.1
10.133, ...
Connexion à raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 10224 (10,0K) [text/plain]
Enregistre : «googletts.agi.3»

googletts.agi.3             100%[=========================================>]   9,98K  --.-KB/s    ds 0,001s

2021-04-21 21:45:15 (17,7 MB/s) - «googletts.agi.3» enregistré [10224/10224]

root@berenger:/var/lib/asterisk/agi-bin# chmod +x googletts.agi
root@berenger:/var/lib/asterisk/agi-bin#
```

Et voila googletts est installé, maintenant voyons comment le faire parler.

Les premiers mots d’Asterisk

La configuration des **IVR** se fait dans le fichier **extensions.conf** Ouvrez votre fichier extensions.conf et copiez les lignes suivantes à la fin du fichier.

```
;Premier IVR
exten => 8000,1,Answer()
exten => 8000,2,agi(googletts.agi,"je m'appelle Berenger oubien deranger et je vais vous montrer comment chante$

exten => 8000,3,Hangup()
```

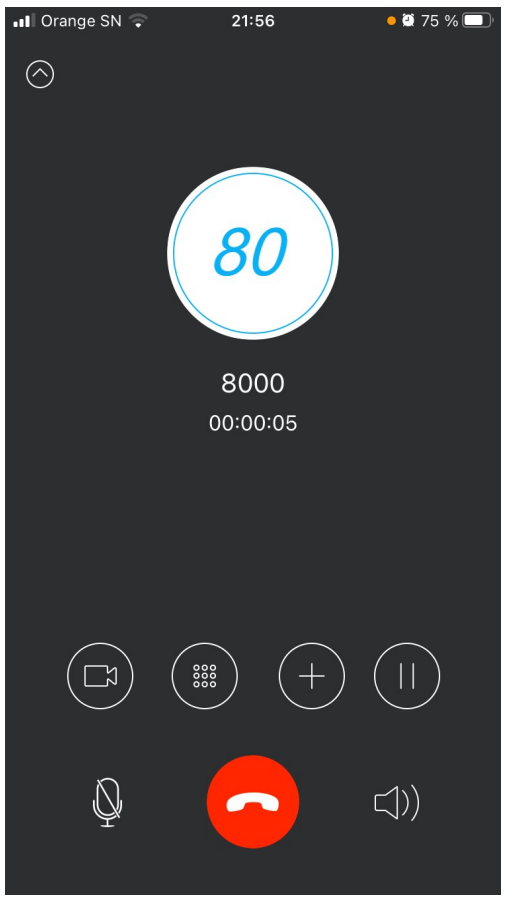
Explications:

exten => 8000,1,Answer() : On crée **l'extensions 8000** pour appeler **l'IVR** et on utilise l'application **Answer()** pour que ça soit **Asterisk** qui réponde étant donné que ce n'est pas un utilisateur que l'on appelle.
exten => 8000,2,agi(googletts.agi, »je m'appelle Berenger oubien deranger et je vais vous montrer comment chanter ...»,fr): Après avoir répondu on utilise donc **googletts** pour dire la phrase suivante : Par Toutatix je parle !, en spécifiant la langue de sortie (fr)
exten => 8000,3,Hangup(): A la fin de la phrase Asterisk raccroche

Maintenant faites un reload et appelez le 8000 à partir de n'importe quel utilisateur pour entendre Asterisk parler.

Un user compose le numéro 8000 voir la capture:

RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)



coté serveur :

```
       > 0x7f99ec0268c0 -- Strict RTP learning after remote address set to: 192.168.100.42:10052
    -- Executing [8000@club:1] Answer("SIP/5000-0000000c", "") in new stack
    -- Executing [8000@club:2] AGI("SIP/5000-0000000c", "googletts.agi,"je m'appelle Berenger oubien deranger et
 je vais vous montrer comment chanter hymne national de mon pays la RCA on y va Ô Centrafrique, ô berceau des Ba
ntous !Reprend ton droit au respect, à la vie ! Longtemps soumis, longtemps brimé par tous,Mais dès ce jour, bri
sons la tyrannie.Dans le travail, l'ordre et la dignité,Tu reconquiers ton droit, ton unité, Et pour franchir ce
tte étape nouvelle,De nos ancêtres la voix nous appelle.Au travail, dans l'ordre et la dignité,Dans le respect d
u droit, dans l'unité,Brisons la misère et la tyrannie,Brandissons l'étendard de la Patrie.",fr") in new stack
    -- Launched AGI Script /var/lib/asterisk/agi-bin/googletts.agi
       > 0x7f99ec0268c0 -- Strict RTP switching to RTP target address 192.168.100.42:10052 as source
    -- <SIP/5000-0000000c> Playing '/tmp/5ab84b16db28a7e658516075e1ac0bef.slin' (escape_digits=) (sample_offset
0) (language 'fr')
       > 0x7f99ec0268c0 -- Strict RTP learning complete - Locking on source address 192.168.100.42:10052
    -- <SIP/5000-0000000c> Playing '/tmp/87e80ed07f6f7d6e69d4d9213776b5a7.slin' (escape_digits=) (sample_offset
0) (language 'fr')
[Apr 21 21:54:31] NOTICE[1628]: chan_sip.c:16041 sip_reg_timeout:     -- Registration for 'serveur(A)@192.168.1.4
' timed out, trying again (Attempt #548)
    -- <SIP/5000-0000000c> Playing '/tmp/89969fbe97f0377dd4b8b4012c2909a9.slin' (escape_digits=) (sample_offset
0) (language 'fr')
    -- <SIP/5000-0000000c> Playing '/tmp/3ec46b2ea8f87a4262abfe5e46b5ef2f.slin' (escape_digits=) (sample_offset
0) (language 'fr')
[Apr 21 21:54:51] NOTICE[1628]: chan_sip.c:16041 sip_reg_timeout:     -- Registration for 'serveur(A)@192.168.1.4
' timed out, trying again (Attempt #549)
    -- <SIP/5000-0000000c>AGI Script googletts.agi completed, returning 0
    -- Executing [8000@club:3] Hangup("SIP/5000-0000000c", "") in new stack
  == Spawn extension (club, 8000, 3) exited non-zero on 'SIP/5000-0000000c'
```

Mettre en place un vrai IVR

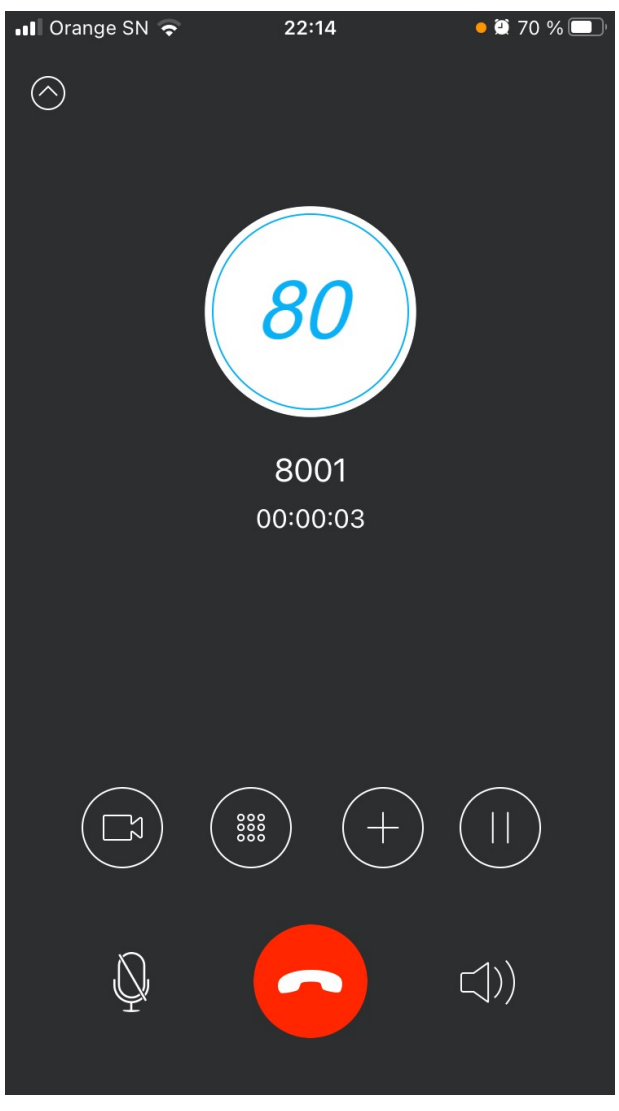
Maintenant nous allons mettre en place un véritable IVR qui va permettre d’appeler nos utilisateurs **Berenger** et **BENAM**.

;Second IVR

```
;Asterisk répond
exten => 8001,1,Answer()
;On met un timeout de 10 secondes pour le choix du destinatire
exten => 8001,2,Set(TIMEOUT(response)=10)
;On annonce les différents choix
exten => 8001,3,agi(googletts.agi,"Bienvenues chez test point com!",fr,any)
exten => 8001,4,agi(googletts.agi,"Qui souhaitez vous joindre?",fr,any)
exten => 8001,5,agi(googletts.agi,"Pour John tapez 1",fr,any)
exten => 8001,6,agi(googletts.agi,"Pour Albert  tapez 2",fr,any)
exten => 8001,7,agi(googletts.agi,"Appuyez sur dièse si vous souhaitez
réécouter ce  message",fr,any)
;On attend que l'utilisateur appuis sur une touche
exten => 8001,8,WaitExten()

;Si l'utilisateur appuis sur 1 on va à la priorité 1 du numéro 6001
exten => 1,1,Goto(6001,1)
;Si l'utilisateur appuis sur 1 on va à la priorité 1 du numéro 6002
exten => 2,1,Goto(6002,1)
;Si l'utilisateur tape un numéro compris entre 3 et 9 et # il retourne à l'étape 3
de l'IVR
exten => _[3-9#],1,Goto(8001,3)
;Si l'utilisateur ne fais rien il retourne à l'étape 3 de l'IVR au bout de 10
secondes.
exten => t,1,Goto(8001,3)
```

Et voila vous pouvez maintenant faire un reload et tester votre IVR.

User 5000 compose 8001 voir la capture :



coté serveur :

```
    -- <SIP/5000-0000000f>AGI Script googletts.agi completed, returning 0
    -- Executing [8001@club:4] AGI("SIP/5000-0000000f", "googletts.agi,"Qui souhaitez vous joindre?",fr,any") in
 new stack
    -- Launched AGI Script /var/lib/asterisk/agi-bin/googletts.agi
    -- <SIP/5000-0000000f> Playing '/tmp/289655073d1909634d11bead581954ff.slin' (escape_digits=0123456789#*) (sa
mple_offset 0) (language 'fr')
    -- <SIP/5000-0000000f>AGI Script googletts.agi completed, returning 0
    -- Executing [8001@club:5] AGI("SIP/5000-0000000f", "googletts.agi,"Pour John  tapez 1",fr,any") in new stac
k
    -- Launched AGI Script /var/lib/asterisk/agi-bin/googletts.agi
    -- <SIP/5000-0000000f> Playing '/tmp/5d3c32755daa4eadc7eb75cd00ae0ae9.slin' (escape_digits=0123456789#*) (sa
mple_offset 0) (language 'fr')
    -- <SIP/5000-0000000f>AGI Script googletts.agi completed, returning 0
    -- Executing [8001@club:6] AGI("SIP/5000-0000000f", "googletts.agi,"Pour Albert tapez 2",fr,any") in new sta
ck
    -- Launched AGI Script /var/lib/asterisk/agi-bin/googletts.agi
    -- <SIP/5000-0000000f> Playing '/tmp/627e7c3625a45b48d1e0c55585593dca.slin' (escape_digits=0123456789#*) (sa
mple_offset 0) (language 'fr')
    -- <SIP/5000-0000000f>AGI Script googletts.agi completed, returning 0
    -- Executing [8001@club:7] AGI("SIP/5000-0000000f", "googletts.agi,"Appuyez sur dièse si vous souhaitez rééc
outer ce  message",fr,any") in new stack
    -- Launched AGI Script /var/lib/asterisk/agi-bin/googletts.agi
    -- <SIP/5000-0000000f> Playing '/tmp/da8add98039dd9c67db56c928141e016.slin' (escape_digits=0123456789#*) (sa
mple_offset 0) (language 'fr')
```

Rendre plus clair le fichier extensions.conf avec des Goto

Nous pouvons rendre plus clair et lisible le fichier **extensions.conf** avec des **Goto** Voici par exemple la même déclaration des **IVR** dans le fichier **extensions.conf** mais avec des **Goto.**

```
;extensions des IVR
exten => 8000,1,Goto(first-ivr,s,1)
exten => 8001,1,Goto(second-ivr,s,1)

;Contexte first-ivr
[first-ivr]
;Premier IVR
exten => s,1,Answer()
exten => s,2,agi(googletts.agi,"Par Berenger je parle !",fr)
exten => s,3,Hangup()

;Contexte second-ivr
[second-ivr]
;Second IVR
exten => s,1,Answer()
exten => s,2,Set(TIMEOUT(response)=10)
exten => s,3,agi(googletts.agi,"Bienvenus chez test point com!",fr,any)
exten => s,4,agi(googletts.agi,"Qui souhaitez vous joindre?",fr,any)
exten => s,5,agi(googletts.agi,"Pour John  tapez 1",fr,any)
exten => s,6,agi(googletts.agi,"Pour Albert  tapez 2",fr,any)
exten => s,7,agi(googletts.agi,"Appuyez sur dièse si vous souhaitez réécouter ce  message",fr,any)
exten => s,8,WaitExten()
```

RAPPORT DE TOIP produit par : BERENGER BENAM (L2TR)

```
exten => 1,1,Goto(work,6001,1)
exten => 2,1,Goto(work,6002,1)
exten => _[3-9#],1,Goto(second-ivr,s,3)
exten => t,1,Goto(second-ivr,s,3)
```