

Déployer un service de téléphonie multi-sites

SAE.3.ROM.04

SAMUEL ROLLI

Table des matières

I.	Appels internes.....	- 2 -
1.	Configuration d'un serveur IPBX.....	- 2 -
2.	Validation des appels.....	- 4 -
3.	Challenge.....	- 4 -
II.	Configuration des appels inter-sites.....	- 5 -
1.	Appels inter-sites.....	- 5 -
2.	Appels externes.....	- 5 -
3.	Challenge.....	- 5 -
III.	Fonctions téléphoniques.....	- 6 -
1.	Transfert d'appel.....	- 6 -
2.	Mise en attente.....	- 7 -
3.	Interception d'appels.....	- 8 -
4.	Enregistrement de conversation.....	- 8 -
5.	Prédécroché.....	- 9 -
IV.	Boîtes vocales.....	- 10 -
1.	Configuration serveur.....	- 10 -
2.	Configuration Clients.....	- 11 -
V.	Conclusion.....	- 12 -

Au cours de cette SAE, notre mission consiste à mettre en place et à configurer de manière approfondie un serveur IPBX Asterisk. Nous opérons au sein d'une maison médicale abritant huit cabinets, chacun composé d'un praticien, d'un assistant et d'une secrétaire, tous dotés de postes IP individuels.

Les praticiens, ou patrons, utilisent le Yealink T31P, les assistants le Nortel LIP6812, tandis que les secrétaires ont recours au softphone Linphone.

L'objectif principal est d'ajuster la configuration de ces téléphones pour permettre à leurs utilisateurs de tirer pleinement parti de leurs fonctionnalités. Nous nous investissons également dans la configuration approfondie des communications internes entre les cabinets, des appels aux patients, et bien sûr, de la possibilité pour ces derniers d'entrer en contact avec les praticiens.

I. Appels internes

1. Configuration d'un serveur IPBX

Après avoir installé Debian 12 sur une machine virtuelle VirtualBox, j'ai suivi la procédure d'installation du site "Wiki ubuntu-fr" (<https://doc.ubuntu-fr.org/asterisk>). Par la suite, j'ai déclaré les différents postes dans le fichier *pjsip_wizard.conf*.

<pre>[patron] type=wizard transport=transport-udp accepts_auth=yes accepts_registrations=yes inbound_auth/type=userpass inbound_auth/username=patron inbound_auth/password=toto aor/max_contacts=1 aor/remove_existing= 1 endpoint/ direct_media=no endpoint/ allow=!all,alaw endpoint/message_context=Messages endpoint/callerid=1</pre>	<pre>[assistant] type=wizard transport=transport-udp accepts_auth=yes accepts_registrations=yes inbound_auth/type=userpass inbound_auth/username=assistant inbound_auth/password=toto aor/max_contacts=1 aor/remove_existing= 1 endpoint/direct_media=no endpoint/allow=!all,alaw endpoint/message_context=Messages endpoint/callerid=2</pre>	<pre>[secretaire] type=wizard transport=transport-udp accepts_auth=yes accepts_registrations=yes inbound_auth/type=userpass inbound_auth/username=secretaire inbound_auth/password=toto aor/max_contacts=1 aor/remove_existing= 1 endpoint/ direct_media=no endpoint/ allow=!all,alaw endpoint/message_context=Messages endpoint/callerid=3</pre>
---	---	---

```
[default]
include => demo
exten => 1,1,Dial(PJSIP/patron)
exten => 2,1,Dial(PJSIP/assistant)
exten => 3,1,Dial(PJSIP/secraitaire)
```

Ensuite, j'ai déclaré le plan de numérotation simple dans le fichier *extensions.conf*

Configuration des téléphones clients

1) Linphone

Pour le softphone Linphone, il faut se rendre dans l'assistant de compte, puis cliquer sur "utiliser un compte sip" puis remplir les champs "nom d'utilisateur" et "nom d'affichage" avec le nom du compte que l'on veut enregistrer (ici "secretaire"), le champ "Domaine SIP" avec l'IP du serveur IPBX (ici 10.129.8.168) et en fin le champ "Mot de passe" y mettre toto.

Pour enregistrer des contacts, il faut se rendre dans la section "CONTACTS" puis cliquer sur le bouton "AJOUTER UN CONTACT" en suite remplir le nom d'affichage par le nom de la personne, enfin ajouter l'adresse SIP sous la forme de <extension>@<ip du serveur IPBX> (exemple, pour le patron, 1@10.129.8.168)

2) Nortel LIP6812

Pour ce téléphone, pas d'interface web ou graphique, il faut naviguer dans les menus de réglages avec les touches actions du téléphone. Tout d'abord, il faut, cliquer sur la touche "Settings" puis à l'aide des flèches naviguer vers le menu "2. SIP Configuration" puis "1. Line 1 Settings". Ensuite, il faut remplir les champs suivants :

- "1. Call Server Address" → 10.129.8.168
- "2. Proxy Port" → 5060
- "3. Display Name" → ASSISTANT
- "4. Name" → assistant
- "5. Auth. Username" → assistant
- "6. Auth. Password" → toto

Pour les touches de *Speed dial*, se rendre dans les paramètres avec la touche physique "Settings", puis naviguer jusqu'à "3. Phone Settings" puis "6. Flexible Key Settings" puis appuyer sur la touche qu'on souhaite paramétrer (les touches de lignes s'affichent en rouges et les touches déjà paramétrées sont en vert), en suite, sélectionner la fonction "Speed Dial Function", enfin, rentrer l'extension que l'on veut joindre.

3) Yealink T31P

Pour le téléphone Yealink, toute la configuration s'est faite sur l'interface web.

Pour enregistrer le téléphone auprès du serveur IPBX, il faut se rendre dans la catégorie "Account" → "Register", en suite, il faut modifier les champs "Account" y mettre "Account1" puis "Line Active" y mettre "Enable" en suite "Label", "Display Name", "Register Name" et "User Name" mettre le nom du compte qu'on veut enregistrer (ici "patron") puis "Password" taper "toto" en fin "Server Host" remplir l'adresse IP du serveur (ici 10.129.8.168).

Pour vérifier que les paramètres renseignés sont justes, le champ "Register Status" passe de "Registering" à "Registered".

Pour les touches d'appel rapide, il faut se rendre dans la rubrique "DSSKey" puis "Line Key" remplir les champs de la manière suivante :

- Type → SpeedDial
- Value → <extension à joindre>
- Label → <nom de la personne à joindre>
- Line → auto

Pour la présence sip, il faut déclarer dans le fichier *extensions.conf* des nouveaux paramètres :

exten => 1, hint, PJSIP/patron, CustomPresence:1
exten => 2, hint, PJSIP/assistant, CustomPresence:2
exten => 3, hint, PJSIP/secretaire, CustomPresence:3

Et modifier les DSSKey, changer le champs Type de "SpeedDial" vers "BLF"

2. Validation des appels

Après avoir tester les appels suivants :

- patron → assistant
- patron → secrétaire
- secrétaire → patron
- secrétaire → assistant
- assistant → patron
- assistant → secrétaire

Tout en ayant une instance de *sngrep* en marche sur le serveur, et en ayant activé l'affichage des flux RTP (touche F3) j'ai pu valider les échanges de signalisation et des flux de transport média bidirectionnels.

3. Challenge

Pour choisir les sonneries en fonction des appelant, il faut se rendre dans l'onglet "*Directory*" puis "*Local Directory*" puis remplir les champs suivants :

- *Name* → <nom de la personne>
- *Office Number* → <extension de la personne>
- *Ring Tone* → <choisir une sonnerie différente des autres>

En fin cliquer sur *Add*.

II. Configuration des appels inter-sites

1. Appels inter-sites

Tout d'abord il faut permettre à notre IPBX de s'identifier au près du serveur opérateur. Pour se faire il faut rajouter un contexte dans le fichier *pjsip_wizard.conf*.

```
[opérateurvoix1]
type=wizard
sends_auth=yes
sends_registrations=yes
remote_host=10.129.10.20
outbound_auth/username=table2
outbound_auth/password=toto
endpoint/from_user=table2
endpoint/from_domain=10.129.10.20
endpoint/allow=alaw
```

Pour vérifier l'identification, il nous faut utiliser la commande *pjsip show registration*, si les informations sont correctes, la ligne "*opérateurvoix1 ...*" indiquera "*Registered*" dans le champ "*Status*".

Pour permettre aux appels de sortir vers le serveur, il m'a fallu ajouter une entrée dans le fichier *extensions.conf* :

```
exten => _*,1,Dial(PJSIP/${EXTEN:1}@opérateurvoix1)
```

Afin de tester ces paramètres, j'ai essayé d'appeler les tables 1, 3 et 4 qui étaient les seuls à être à ce niveau d'avancement au moment où j'avais besoin de tester.

2. Appels externes

Pour permettre aux numéros à dix chiffres de sortir, j'ai ajouté cette ligne à mon fichier *extensions.conf* :

```
exten => _ONXXXXXX.,1,Dial(PJSIP/${EXTEN}@opérateurvoix1)
```

Et pour tester, j'ai contacté le téléphone TM2 (le TM1 était en panne) au 0212345678.

3. Challenge

Le challenge de messagerie instantanée inter-sites n'a pas été réalisé. Par manque de temps et par des recherches pour d'autres parties qui se sont éternisées et ont donc compromis la mise en place de cette fonctionnalité.

III. Fonctions téléphoniques

1. Transfert d'appel

1) Activation du transfert d'appel

Activer le transfert d'appel est une chose simple. Il suffit d'ajouter les option "Tt" dans le champ option de la fonction *Dial()* puis exécuter la commande *dialplan reload* dans le CLI d'asterisk ou faire un redémarrage complet en exécutant la commande *core stop now* dans le CLI puis *asterisk -cvvvvv* dans le terminal pour relancer le serveur.

exten=>1,1,Dial(PJSIP/patron,,Tt)	Exten=>2,1,Dial(PJSIP/assistant,,Tt)	Exten=>3,1,Dial(PJSIP/secretaire,,Tt)
-----------------------------------	--------------------------------------	---------------------------------------

On peut voir ici les deux virgules qui indique de l'on ignore le champ "timer" qui est sensé indiquer combien de temps le téléphone doit sonner avant de passer a l'instruction suivante, pour passer au champs option ou l'on indique la possibilité d'être transférer et de transférer.

2) Touche de fonction rapide

Après m'être rendu compte que les touches natives au Nortel ne m'aurait pas permis de faire les fonctions dont j'avais besoin. J'ai suivi vos indications et je me suis penché vers les fonction dynamiques et utiliser la fonction "*Transfer()*". Il m'a donc fallu ajouter dans le fichier *features.conf* les lignes suivantes :

```
[applicationmap]
Tto1 = 1,peer,Transfer,PJSIP/patron
Tto2 = 2,peer,Transfer,PJSIP/assistant
Tto3 = 3,peer,Transfer,PJSIP/secretaire
```

"Tto1" est le nom de ma fonction ("T" pour transfert "to" pour indiquer vers et "1" pour l'extensions contactée), en suite "1" indique la touche DTMF qui déclenchera la fonction, "peer" indique le la fonction est à exécuter sur le correspondant distant, "Transfer" indique la fonction que le serveur devra exécuter, et enfin, "PJSIP/patron" indique la destination du transfert.

Pour permettre aux différents téléphones de les exécuter, il faut leur en donner la permission. Dans le fichier *extensions.conf* au début de chaque scénario d'appel, il faut ajouter cette ligne (cette information est donnée dans les exemples du fichier *features.conf*) :

```
exten => 1, 1, Set(__DYNAMIC_FEATURES=Tto1#Tto2#Tto3)
```

Ici la fonction *Set()* prends en argument "__DYNAMIC_FEATURES" qui indique que les fonctions sont à trouver dans le contexte "*applicationmap*" dans le fichier *features.conf*, puis il suffit de lister les fonctions que l'on veut ajouter séparées par un "#".

Après avoir relancer le serveur, j'ai pu tester les scénarii possibles avec les touches de transfert 1 pour transférer vers le patron, 2 vers l'assistant et 3 vers la secrétaire.

2. Mise en attente

1) Trouver une musique d'attente personnalisée

Pour avoir le message d'attente le plus personnalisé possible, j'ai décidé de l'enregistrer moi-même. J'ai donc ajouté dans le fichier *extensions.conf* la ligne suivante :

```
exten => 560, 1, Record( /var/lib/asterisk/sounds/MessageVocaux/Temp.gsm)
```

L'ajout de cette ligne me permet d'enregistrer un message vocal, en appelant le numéro 560, en disant mon message et en appuyant sur la touche "#", le message est enregistré dans l'IPBX.

Ensuite, j'ai dû le récupérer sur ma machine physique car le son n'était pas assez fort. J'ai donc utilisé les commandes *scp* pour faire une copie du fichier avec SSH sur ma machine physique, puis à l'aide de convertisseur *gsm* → *mp3* je l'ai converti, ensuite, j'ai utilisé l'application *Audacity* pour augmenter artificiellement le niveau sonore du message.

Ensuite, j'ai exécuté les commandes suivantes :

```
mkdir /var/lib/asterisk/moh_rolli
scp toto@10.129.8.12:Tmp.mp3 /var/lib/asterisk/moh_rolli/cab_lib_rolli.mp3
```

Qui m'ont permis de créer le dossier *moh_rolli* et de copier le fichier *Tmp.mp3* (qui était dans le dossier */home/toto* de la machine hôte) dans le bon dossier tout en le renommant au passage.

En suite dans le fichier *musiconhold.conf* j'ai ajouter ces lignes :

```
[default]
mode=custom
directory=/var/lib/asterisk/moh_rolli
application=/usr/bin/madplay -a -12 --mono -R 800 -output=raw:-
```

Elles me permettent de définir un dossier dans lequel lire les musiques avec l'option *directory* et de définir une application pour lire ces fichiers, car asterisk n'est pas capable de lire les fichiers *.mp3* nativement (sans qu'on ne lui ai installer le module qui en est capable).

Enfin, j'ai installé l'application *madplay* avec la commande *apt install madplay*.

2) Touche rapide de mise en attente

Pour mettre les appels en attente, asterisk a une fonction native qui s'appelle "*parkedcalls*" (littéralement un parking pour les appels) et qui activée par la combinaison #700 pour l'activer il suffit d'ajouter la ligne "*include => parkedcalls*" dans le contexte "*[default]*" du fichier *extensions.conf*.

Pour paramétrer une touche rapide (du même type que la touche de transfert) il m'a fallut me rendre dans le fichier *features.conf* :

```
[featuremap]
parkcall => 9
```

Ce qui me permet d'activer la fonction de "parking d'appel" sur la touche DTMF 9. Ensuite, il ma fallu ajouter l'option "*Kk*" dans la fonction *dial* de chaque scénario d'appel, pour permettre à tous mes utilisateurs d'utiliser cette fonction :

```
exten=>1,1,Dial(PJSIP/patron,,TtKk)
```

```
exten=>2,1,Dial(PJSIP/assistant,,TtKk)
```

```
exten=>3,1,Dial(PJSIP/secretaire,,TtKk)
```

3. Interception d'appels

1) Définir un groupe d'appel

Pour définir un groupe d'appel et d'interception, il faut ajouter dans le fichier *pjsip_wizard.conf* les lignes suivantes sur les participants au groupe (ici secretaire et assistant):

```
endpoint/call_group=1  
endpoint/pickup_group=1
```

2) Tester

Ensuite, pour tester, il suffit d'appeler l'un des membres du groupe et de numéroté "*"8" sur le second membre. Si l'appel est redirigé automatiquement, la fonction est implémentée correctement.

3) Touche de fonction sur le Nortel

Pour cette fonction, les touches paramétrables du Nortel sont compatibles, j'ai donc implémenté une touche type "*Speed Dial*" vers l'extension "*"8".

Pour se faire il faut se rendre dans les paramètres avec la touche physique "*Settings*", puis naviguer jusqu'à "*3. Phone Settings*" puis "*6. Flexible Key Settings*" puis appuyer sur la touche qu'on souhaite paramétrer (les touches de lignes s'affichent en rouges et les touches déjà paramétrées sont en vert), en suite, sélectionner la fonction "*Speed Dial Function*", enfin, rentrer l'extension "*"8".

4. Enregistrement de conversation

Sur cette version de Asterisk, l'enregistrement des conversations est instable. J'ai tout de même faire une configuration qui permettra d'enregistrer ces conversations.

1) Activation coté serveur

Pour activer l'enregistrement de conversations, il faut modifier une ligne du fichier *features.conf*:

```
automon => *1
```

Par défaut elle a un point-virgule devant, ce qui est la marque du commentaire sur asterisk. Simplement le retirer, active l'enregistrement coté serveur.

Ensuite, il faut autoriser les utilisateurs à utiliser cette fonction. Pour ce faire, il faut ajouter les option W et w dans la fonction *Dial* du scénario d'appel :

```
exten=>1,2,Dial(PJSIP/Patron,,TtKkWw)
```

```
exten=>2,2,Dial(PJSIP/assistant,,TtKkWw)
```

```
exten=>3,2,Dial(PJSIP/secretaire,,TtKkWw)
```

2) Tester

Pour tester l'implémentation, j'ai effectué un appel, puis j'ai lancer un enregistrement. J'ai vu sur le CLI que l'enregistrement était lancer. J'en ai déduit que la configuration était bonne.

3) Touche de fonction

Ici j'ai juste modifié la touche d'activation de cette fonction, je l'ai passé de *1 à 8. N'importe quel téléphone où les options Ww sont activées, le fait de presser la touche DTMF 8 lancera l'enregistrement de la conversation.

5. Prédécroché

1) Implémenter un message de prédécroché

Pour le prédécroché, j'ai tout d'abord enregistré un message personnalisé à l'aide de la fonction *record* que j'avais mis sur l'extension 560.

Ensuite, j'ai exécuté les commandes suivantes :

```
mkdir /var/lib/asterisk/sounds/predec_rolli/
mv /var/lib/asterisk/sounds/MessageVocaux/Tmp.gsm \
/var/lib/asterisk/sounds/predec_rolli/predec.gsm
```

En suite j'ai utilisé les fonctions de *playback* sur les scénarii d'appel :

```
exten => 1,2,playback(predec_rolli/predec)
```

Pour tester, j'ai lancé un appel local, et j'ai attendu la fin du message de prédécroché, l'appel c'est lancé, ce qui m'a indiqué que les fonctions *Dial* se sont bien exécutées et donc que mon prédécroché fonctionnait correctement.

2) Activer le prédécroché uniquement sur le faisceau SIP

Cette fonction et celle qui a été le plus chronophage en termes de temps de recherche et de lecture de documentation. J'ai décidé d'utiliser la fonction *GotoIf* qui permet de se déplacer vers une priorité ou un autre contexte si un test est validé. Je n'ai peut-être pas choisi la manière la plus simple de le faire, et je sais que cette méthode fonctionne uniquement car je n'ai pas beaucoup d'utilisateur interne. Si j'avais par exemple 40 utilisateurs internes, il serait difficile de faire le test pour les 40 utilisateurs internes.

Syntaxe de la fonction *GotoIf* : *GotoIf(<test>?<priorité si validé>:<priorité si pas validé>)*

```
exten => 1,2,GotoIf(["${CALLERID(num)}"="1" | \
"${CALLERID(num)}"="2" | "${CALLERID(num)}"="3"]?4:3)
exten => 1,3,playback(predec_rolli/predec)
exten => 1,4,Dial(PJSIP/patron,,TtWwKk)
```

Explication :

"*[\$]*" indique que l'on va tester tout ce qui est dans les crochets.

"*{}*" indique que ce qui est à l'intérieure est une variable.

"*CALLERID(num)*" est la variable qui contient le numéro de la personne qui appelle. Le fait que *s{CALLERID(num)}* soit entre guillemet indique que l'on fait une comparaison de valeur et pas de type de variable.

"*|*" symbole qui signifie "ou" dans la configuration asterisk

"*?4:3*" indique que le test est fini. Si le test est validé, il faut aller directement à la priorité 4 de la configuration, dans le cas contraire, se rendre à la priorité 3.

IV. Boîtes vocales

1. Configuration serveur

Pour configurer les boîtes vocales, je me suis rendu dans le fichier *voicemail.conf*, et j'ai ajouté les lignes suivantes dans le contexte *[default]* :

```
1 => 1234, boîte vocale du patron, ipbx@10.129.8.168
2 => 1234, boîte vocale de l'assistant, ipbx@10.129.8.168
4 => 1234, boîte vocale de la secrétaire, ipbx@10.129.8.168
```

Ensuite, j'ai ajouté la ligne "*aor/mailboxes=<extension de l'utilisateur>@default*" à chaque utilisateur dans le fichier de configuration *pjsip_wizard.conf*.

Cette configuration crée une boîte vocale par utilisateur avec comme mot de passe 1234.

1) MWI sur le Yealink

Se rendre sur l'interface de configuration web. Puis dans la rubrique "Account" puis "Advanced" et modifier les champs suivants :

Subscribe for MWI → *Enabled*

MWI Subscription period → *3600*

Voice mail → *1*

Voice mail display → *enabled*

Ensuite, il faut redémarrer le téléphone Yealink, débrancher et rebrancher le câble fonctionne très bien.

Pour vérifier que le téléphone est bien abonné à la MWI, on peut utiliser la commande "*pjsip show subscription inbound*" et vérifier qu'une troisième entrée est présente. On peut aussi tester en laissant un message au téléphone et voir si le message "*1 new voice message*" apparaît sur le téléphone.

2) Ajouter le timer de basculement

Dans le fichier *extensions.conf*, il faut ajouter "10" dans les fonctions *Dial* :

```
exten=>1,2,Dial(PJSIP/patron,10,TtKkWw)
exten=>2,2,Dial(PJSIP/assistant,10,TtKkWw)
exten=>3,2,Dial(PJSIP/secretaire,10,TtKkWw)
```

3) Numéro de consultation directe

Pour faire un numéro de consultation directe il faut ajouter dans le fichier de configuration *extensions.conf* la ligne suivante :

```
exten => <n°>,1,VoiceMailMain(<id BV>@default)
```

Je trouvais ça pas

n'importe qui pouvait consulter les boîtes mails de n'importe qui à conditions qu'ils aient le mot de passe. J'ai rendu l'accès vocal : très sécurisé que donc décidé de automatique aux boîtes

```
exten => 888,1,VoiceMailMain(${CALLERID(num)}@default)
```

Maintenant, l'accès à toutes les boîtes vocales se fait à l'aide du 888 et l'identifiant de la boîte vocale est rempli automatiquement à l'aide du "\$CALLERID(num)".

Ensuite, je ne trouvais pas sécurisé le fait que l'on puisse accéder au 888 depuis le faisceau SIP. J'ai donc ajouté un test avec un message personnalisé dans le cas où le numéro viendrait du faisceau SIP.

```
exten => 888,1,GotoIf("${CALLERID(num)}"="1" | \
"${CALLERID(num)}"="2" | "${CALLERID(num)}"="3"?2:10)
exten => 888,2,VoiceMailMain(${CALLERID(num)}@default)
exten => 888,3,Hangup
exten => 888,10,Answer
exten => 888,11,Playback(voicemail_access/denied)
exten => 888,12,Hangup
```

2. Configuration Clients

1) Nortel

Pour cette fonction, les touches paramétrables du Nortel sont compatibles, j'ai donc implémenté une touche type "*Speed Dial*" vers l'extension "888".

Pour se faire il faut se rendre dans les paramètres avec la touche physique "*Settings*", puis naviguer jusqu'à "3. Phone Settings" puis "6. Flexible Key Settings" puis appuyer sur la touche qu'on souhaite paramétrer (les touches de lignes s'affichent en rouges et les touches déjà paramétrées sont en vert), en suite, sélectionner la fonction "*Speed Dial Function*", enfin, rentrer l'extension "888".

2) Yealink

Pour permettre au yealink de se connecter rapidement à la messagerie, j'avais paramétrer la touche message en *SpeedDial* sur l'extension 888.

Pour ce faire, il faut se rendre dans l'onglet "*Dsskey*" puis dans la rubrique "*Programmable Key*", enfin, il faut modifier la ligne "Message" en mettant le champ "Type" sur "*SpeedDial*", "Line" sur "Auto" et le champ "Value" sur "888".

V. Conclusion

En conclusion, ce projet de déploiement et de configuration avancée du serveur IPBX Asterisk a constitué une expérience intéressante pour moi. J'ai eu l'opportunité de mettre en pratique les compétences acquises tout au long de ma formation, démontrant ainsi ma capacité à appliquer des connaissances théoriques dans des situations concrètes.

L'utilisation approfondie de la documentation a joué un rôle clé dans mon approche, me permettant de résoudre des défis spécifiques rencontrés lors de la configuration. Cette démarche a renforcé ma capacité à assimiler de nouvelles informations et à les appliquer de manière pertinente dans un contexte professionnel.

Malgré mes efforts soutenus, certaines fonctionnalités envisagées n'ont pas pu être pleinement implémentées. Par exemple, l'envoi des messages vocaux par mail, la gestion de la présence SIP sur Linphone, et l'objectif 5 lié au menu vocal interactif n'ont pas pu être réalisés en raison de contraintes de temps et d'expertise. Ces éléments, bien que prévus dans le cahier des charges initial, ont été impactés par des contraintes opérationnelles et la nécessité de prioriser certaines tâches cruciales.

Il est essentiel de souligner que cette expérience m'a offert une compréhension approfondie des défis pratiques liés à la mise en œuvre de systèmes complexes, tout en identifiant des axes d'amélioration pour des projets futurs.