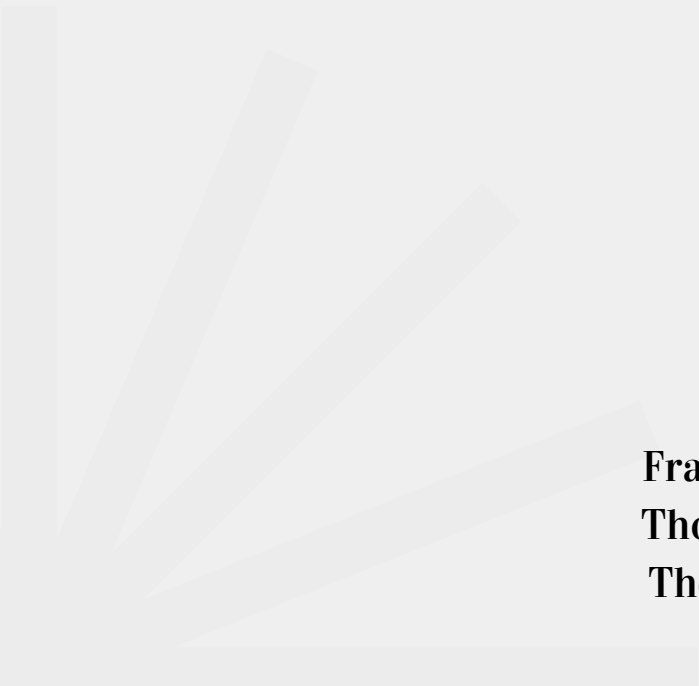




VOIP



**Fraoucene Fares
Thomas Chardin
Thomas Cerruti**

Contexte

Peut-on dire la même chose de ses coûts opérationnels et de maintenance ?

Les coûts opérationnels et de maintenance pour la VoIP dépendent de la taille de l'entreprise, de la complexité du système, des besoins en support et de la bande passante nécessaire. Les coûts incluent les frais mensuels de service, la bande passante, la gestion du personnel et les mises à jour logicielles/matérielles. Bien que la VoIP puisse offrir des économies par rapport aux systèmes traditionnels, il est essentiel d'évaluer tous ces coûts pour en mesurer l'impact financier.

Contexte

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

La configuration VoIP d'un centre d'appels diffère de celle d'un standard téléphonique d'entreprise en raison du volume élevé d'appels, des fonctionnalités spécifiques telles que l'ACD et les files d'attente, de l'intégration avec d'autres systèmes comme les CRM, et de la gestion des agents distants. Les centres d'appels nécessitent une configuration plus complexe pour gérer efficacement ces aspects, tandis que les standards téléphoniques d'entreprise sont souvent plus simples et axés sur les communications internes et externes de base.

Contexte

Identifiez des sites marchands ou de service dont customer service implique des services VoIP, donnez quelques exemples et décrivez une architecture possible de leur système.

Voici une architecture possible pour le service client utilisant la VoIP :

- ****Infrastructure VoIP**** : Utilisation d'un fournisseur VoIP pour gérer les appels.
- ****Intégration CRM**** : Accès aux informations clients pour une assistance personnalisée.
- ****Serveur de routage d'appels**** : Acheminement des appels vers les agents disponibles.
- ****Softphones/Téléphones IP**** : Utilisation d'applications ou de téléphones IP pour les appels.
- ****Enregistrement des appels**** : Pour la formation et le suivi de la qualité.
- ****Analyse des performances**** : Surveillance des performances des agents et des métriques clés.

Cette architecture offre un support client efficace et intégré grâce à la VoIP.

Contexte

Effectuez quelques recherches sur les chiffrements les mieux adaptés à la VoIP.

Les chiffrements les mieux adaptés à la VoIP incluent :

- ****SRTP**** : Pour le chiffrement des flux de médias en temps réel.
- ****TLS et DTLS**** : Pour sécuriser les échanges de données de signalisation.
- ****ZRTP**** : Pour sécuriser les sessions de communication en temps réel.
- ****SDS**** : Pour négocier et établir les clés de chiffrement avec SRTP.

Ces chiffrements offrent une sécurité adéquate tout en minimisant l'impact sur la qualité des appels VoIP.

Projet

Téléchargement Asterisk

```
apt update
apt upgrade
apt install nano wget tar curl
systemctl stop apparmor
apt remove apparmor
cd /usr/src
wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20.7.0.tar.gz
tar zxvf asterisk-20.7.0.tar.gz
rm -rf asterisk-20.7.0.tar.gz
cd asterisk-20.7.0/
contrib/scripts/install_prereq install
./configure
make menuselect
make
contrib/scripts/get_mp3_source.sh
make install
make samples
mkdir /etc/asterisk/samples
mv /etc/asterisk/*.*/etc/asterisk/samples/
make basic-pbx
make config
systemctl enable asterisk.service
systemctl start asterisk.service
systemctl status asterisk.service
```

ici nous allons d'abord mettre a jour notre vm, on va aussi installer le fichier tar de asterisk disponible sur leur site internet. On le décompresse et on commence a setup asterisk a setup asterisk puis le demarer et verifier si il est fonctionnel

Projet

Configuration Asterisk

```
nano /etc/default/asterisk
  AST_USER="asterisk"
  AST_GROUP="asterisk"
sudo nano /etc/asterisk/asterisk.conf
  runuser = asterisk           ; The user to run as.
  rungroup = asterisk          ; The group to run as.
sudo systemctl restart asterisk
systemctl status asterisk
```

```
fares@debian:/etc/asterisk$ cat pjsip.conf
;===== TRANSPORTS =====
; Our primary transport definition for UDP communication behind NAT.
[transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

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

[auth_userpass](!)
type=auth
auth_type=userpass

[aor_dynamic](!)
type=aor
max_contacts=1

[fares](endpoint_internal)
auth=fares
aors=fares
[fares](auth_userpass)
password=fares2605
username=fares
[fares](aor_dynamic)

[thomas](endpoint_internal)
auth=thomas
aors=thomas
[thomas](auth_userpass)
password=thomas2605
username=thomas
[thomas](aor_dynamic)
```

Alors dans ce fichier nous allons configurer tous ce qui touche a l'authentification, ici on a créer 2 utilisateurs (Fares et Thomas) avec leur mot de passe respectif et leur identifiant

Projet

Configuration Asterisk (suite)

```
fares@debian:/etc/asterisk$ cat asterisk.conf
[options]
; If we want to start Asterisk with a default verbosity for the verbose
; or debug logger channel types, then we use these settings (by default
; they are disabled).
;verbose = 5
;debug = 2

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

defaultlanguage = fr
```

Ici on va pouvoir configurer la langue utiliser
utiliser par asterisk, par exemple pour la
messagerie et aussi le groupe pouvant utiliser
asterisk (dans le pc).

Projet

Configuration Asterisk (suite)

```
fares@debian:/etc/asterisk$ cat voicemail.conf
[general]
format=wav49|gsm|wav|ulaw
maxmsg=30                ; Max messages
maxsecs=0                ; Durée max message
minsecs=1                ; Durée min message
maxlogins=3              ; Nombre tentative login
review=yes               ; Permet à l'appelant de réécouter son message

[default]
; Numéro de messagerie => mot de passe, nom d'utilisateur
103 => fares2605, fares
102 => thomas2605, thomas _
```

Ici on va pouvoir configurer les différents paramètres de la boîte mail

```
fares@debian:/etc/asterisk$ cat extensions.conf
[from-internal]
exten=>103,1,Dial(PJSIP/fares,10)
exten=>102,1,Dial(PJSIP/thomas,10)
exten=>199,1,VoiceMailMain()      ; Numéro de téléphone du répondeur

; Règles 2
exten=>103,2,VoiceMail(103)       ; Appel répondeur compte 103
exten=>102,2,VoiceMail(102)       ; Appel répondeur compte 102
```

Ici on va pouvoir configurer les raccourcis pour les appels et configurer les boîtes mails si l'interlocuteur ne réponds pas

Projet

Ajout d'une boite vocal TTS

```
apt-get install perl libwww-perl sox mpg123
cd /var/lib/asterisk/agi-bin
wget https://raw.githubusercontent.com/zaf/asterisk-googleletts/master/googleletts.agi
chmod +x googleletts.agi
```

```
exten => 8001,1,Answer()
exten => 8001,2,Set(TIMEOUT(response)=10)
exten => 8001,3,agi(googletts.agi,"Bienvenues dans le serveur asterisk",fr)
exten => 8001,5,agi(googletts.agi,"Pour l'accueil tapez 1",fr,any)
exten => 8001,6,agi(googletts.agi,"Pour le service administration tapez 2",fr,)
exten => 8001,7,agi(googletts.agi,"Appuyez sur dièse si vous souhaitez réécouter ce message",fr)
exten => 8001,8,WaitExten()
exten => 1,1,Goto(102,1)
exten => 2,1,Goto(103,1)
exten => _[3-9#],1,Goto(8001,3)
exten => t,1,Goto(8001,3)
```

Grace a cette configuration, on peut appeler le numéro 8001 et tomber directement sur la messagerie automatisé. Grace a l'api google nous entendrons une voix qui va nous lire les chaines de caractère si dessus, en suivant les étapes dit pas l'api google on va pouvoir soit contacter les service d'accueil (Fares) ou le service administratif (Thomas)

Projet

Installation 3CX

[Lien 3cx](#)

Installation facile & rapide

- ✓ Windows, un opérateur VoIP, Internet et un casque.
- ✓ Compatible avec une large gamme d'opérateurs VoIP / trunks SIP.
- ✓ Supporte les casques standards – pas besoin d'équipement cher.
- ✓ Pour démarrer, téléchargez le [softphone gratuit ici](#).



Projet

Configuration d'un utilisateur

Account settings

Account name: fares

Caller ID: fares

Credentials

Enter your SIP account credentials

Extension: fares

ID: fares

Password: *****

My location

Specify the IP of your PBX/SIP server

☒ I am in the office - local IP 10.10.4.17 of PBX

☐ I am out of the office - external IP of PBX

☐ Use 3CX Tunnel

Eliminates firewall configuration. Requires 3CX Phone System for Windows

Local IP of remote PBX: 192.168.240.128

Tunnel password: *** Port: 5090

☐ Use Outbound Proxy server

Required by some VoIP Providers. Specify IP or name.

☐ Perform provisioning from following URL:

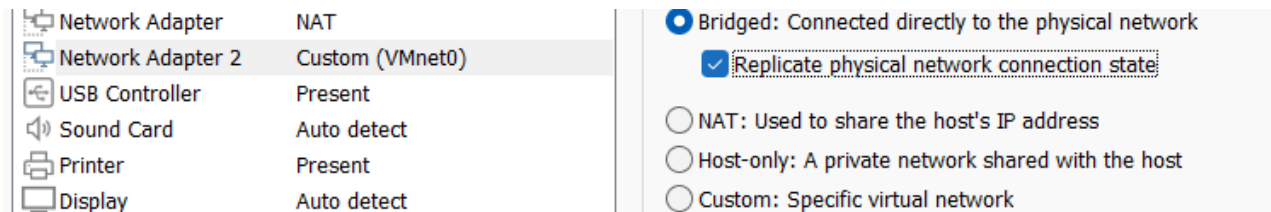
http://

Advanced settings OK Cancel

Après l'installation de 3cx on va pouvoir utiliser un utilisateur créer préalablement sur notre pjsip.conf, on renseigne les différentes infos de l'utilisateur et l'adresse ip de la machine ou est créer asterisk

Projet

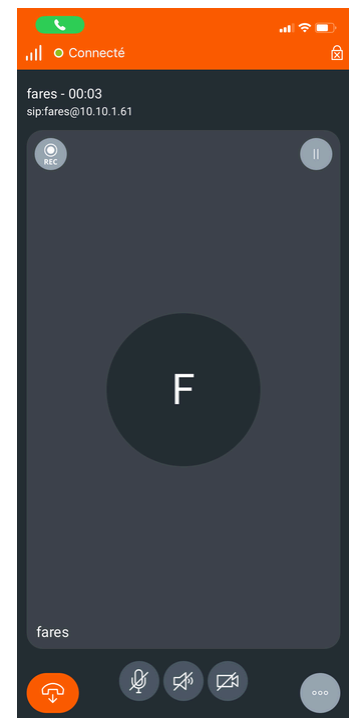
Utilisation de asterisk sur un autre appareil



```
valid_crl forever preferred_crl forever  
3: ens36: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g  
roup default qlen 1000  
    link/ether 00:0c:29:84:69:0f brd ff:ff:ff:ff:ff:ff  
    altname enp2s4  
    inet 10.10.1.61/16 brd 10.10.255.255 scope global dynamic noprefixroute ens3
```



Ici en changeant le type on va pouvoir connecter un appareil annexe (Iphone) a notre serveur asterisk



Projet Intégration

```
sudo apt-get install ldap-utils
URI ldap:172.16.0.250
BASE dc=ADlplateforme,dc=com
ldapsearch -x -H ldap://adresse_IP_AD -b "dc=ADlplateforme,dc=com" -D "ArthurNeutron@adlplateforme.com"
-W
```

```
import csv
from pyad import *

def ajouter_utilisateur(nom, numero_telephone, mdp):
    # Créer un nouvel utilisateur dans AD
    new_user = pyad.aduser.ADUser.create(nom)
    new_user.set_password("1234") # Définir le mot de passe
    new_user.update_attribute("telephoneNumber", numero_telephone) # Définir le numéro de téléphone
    new_user.update_attribute("userAccountControl", pyad.adconstants.ADS_UF_NORMAL_ACCOUNT) #
    Activer le compte

def main():
    with open('utilisateurs.csv', newline='') as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            nom = row['Nom']
            numero_telephone = row['Numéro de téléphone']
            ajouter_utilisateur(nom, numero_telephone)

if __name__ == "__main__":
    pyad.set_defaults(ldap_server="ldap://ADplateforme") # Définir le serveur AD
    main()
```

Projet

Déploiement automatisé

```
# Vérifier les privilèges d'administration
if [[ $(id -u) -ne 0 ]]; then
    echo "Vous n'avez pas les droit administrateur."
    exit 1
fi

apt update
apt upgrade -y

# Dépendances
apt install -y wget build-essential libssl-dev libncurses5-dev libnewt-dev libxml2-dev libsqlite3-dev
uuid-dev

# Téléchargement Asterisk
cd /usr/src
wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-20.7.0.tar.gz
tar xvf asterisk-20.7.0.tar.gz
rm asterisk-20.7.0.tar.gz
cd asterisk-20.7.0/

contrib/scripts/install_prereq install
./configure
make menuselect
make
contrib/scripts/get_mp3_source.sh
make install
make samples
mkdir /etc/asterisk/samples
mv /etc/asterisk/*.*/etc/asterisk/samples/
make basic-pbx
make config

systemctl start asterisk
systemctl enable asterisk

echo "Installation d'Asterisk terminée."
```

Projet

Déploiement automatisé (suite)

Existe-t-il un moyen de standardiser l'installation complète sous forme de containers ? Détailler

Oui, il existe des moyens de standardiser l'installation complète sous forme de containers, et la méthode la plus populaire est l'utilisation de technologies de conteneurisation comme Docker ou Kubernetes. Nous pourrions utiliser Docker pour standardiser mais pour gérer des conteneurs à plus grande échelle nous utiliserons Kubernetes par exemple.