



# Détermination des composantes

Conception d'un logiciel de softphone

Version 1

Martin DELOR

Lauriane LE PAPE

Manon MICHELET

28/10/19

## Historique des révisions

Date	Version	Description	Auteur
31/10/19	Version 1	Première version des composants	TriFrogz

## Définitions

Terme	Définition
Téléphonie IP	Téléphonie faite grâce à notre accès Internet plutôt qu'à une ligne terrestre.
Serveur PBX	Autocommutateur téléphonique privé se servant d'Internet pour la gestion des appels téléphoniques d'une entreprise.
Scrum	Framework lié aux méthodes agiles de gestion de projet.

## Abréviations/acronymes

Abré./Acro.	Définition
PC	Personal Computer
SIP	Session Initiation Protocol
IAX	Inter-Asterisk eXchange
PBX	Private Branch eXchange
FAI	Fournisseur d'Accès à Internet

## Table des matières

Historique des révisions	2
Définitions	2
Abréviations/acronymes	2
Table des matières	3
1. Introduction	4
1.1. Objectifs	4
1.2. Portée	4
1.3. Références	4
2. Description des composants logiciels	5
2.1. Composant 1	5
2.2. Composant 2	5

# 1. Introduction

## 1.1. Objectifs

Ce projet consiste à concevoir un logiciel de softphone afin de pouvoir utiliser le serveur mis en place dans la première partie du projet. Ce logiciel sera conçu à partir de plusieurs langage de programmation tels que Python ou C++ et ouvrira lui-même la communication SIP avec le serveur à l'aide d'un ou plusieurs modules déjà existants. Le but de ce logiciel est de remplir les fonctions demandées dans la partie 1 du projet, c'est-à-dire les boîtes vocales, le renvoi-d'appels et les appels en attente.

## 1.2. Portée

Le produit est un sous-système permettant d'utiliser le serveur installé dans la première partie du projet. Nous devons donc concevoir un logiciel de softphone à l'aide de nos connaissances en programmation mais aussi avec l'aide de modules pré-existant afin de le rendre le plus simple d'utilisation possible. Nous devons donc intégrer toutes fonctions demandées dans la première partie du projet telles que la boîte vocale, la communication avec les télécopieurs et le renvoi d'appel.

## 1.3. Références

softphone librairies / software :

<https://www.voip-info.org/open-source-voip-software/>

doc sur le protocole SIP :

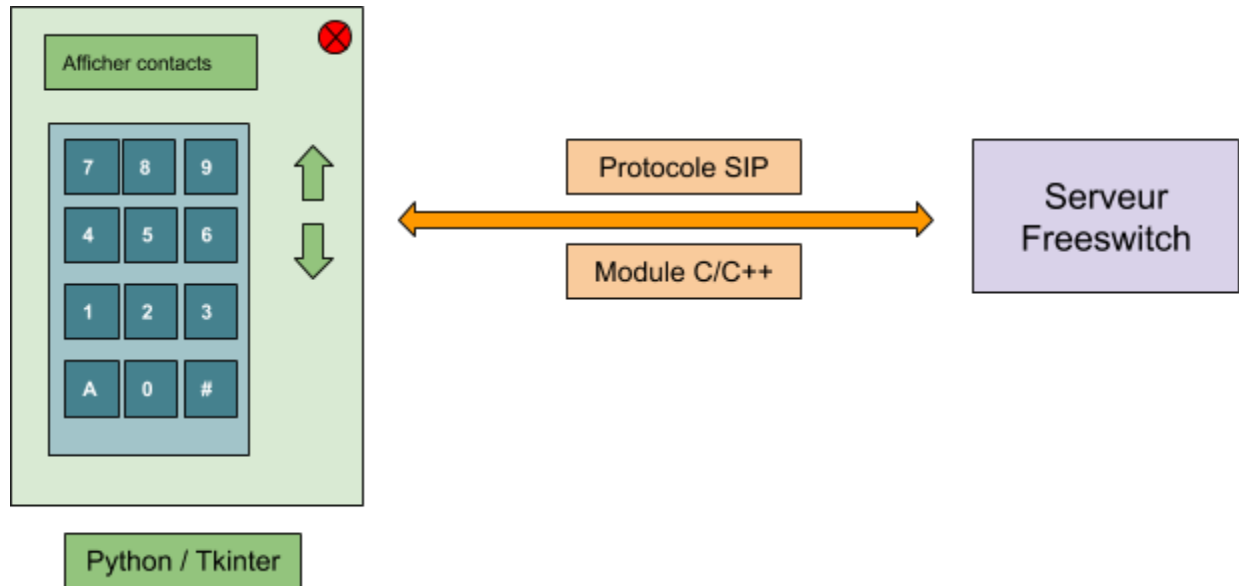
[https://fr.wikipedia.org/wiki/Session\\_Initiation\\_Protocol](https://fr.wikipedia.org/wiki/Session_Initiation_Protocol)

<https://www.3cx.fr/blog/guide-protocole-sip/>

Norme ISO 29110

## 2. Description des composants logiciels

Le système sera composé de 2 parties communicantes : le softphone développé en Python et le serveur Freeswitch. Ci-dessous, nous organisons le détail des composants en fonction de chaque Sprint.



### 2.1. Composant 1 : le softphone

#### *Sprint 1 :*

Protocole : Nous pensons choisir SIP car c'est celui sur lequel nous avons travaillé avec notre projet 1.

Librairie : Après recherches, il semble que la librairie PJSIP puisse convenir pour créer le softphone. A confirmer.

#### *Sprint 2 :*

Protocole : SIP car le choix de notre librairie convient au professeur.

Librairie : Choix d'utiliser la librairie PJSIP et plus précisément l'API PJSUA compatible avec python et C++. Ce choix a été motivé par le fait que c'est une librairie très populaire dans la téléphonie basée sur le protocole SIP et que notre professeur nous l'ait recommandé. [Documentation PJSUA](#)

```

C:\ Command Prompt - pjsua_vc6d --config-file colinux1-tls.cfg
>>>>
Account list:
[ 0] <sip:192.168.0.1:4426;transport=UDP>: does not register
    Online status: Online
[ 1] <sip:192.168.0.1:4426;transport=TCP>: does not register
    Online status: Online
[ 2] <sip:192.168.0.1:4427;transport=TLS>: does not register
    Online status: Online
*[ 3] sip:bulukucing1@colinux: 200/OK (expires=288)
    Online status: Online
Buddy list:
-none-

+-----+-----+-----+
| Call Commands: | Buddy, IM & Presence: | Account: |
+-----+-----+-----+
| m Make new call | +b Add new buddy      | +a Add new acct. |
| M Make multiple calls | -b Delete buddy      | -a Delete acct.  |
| a Answer call      | i Send IM            | !a Modify acct.  |
| h Hangup call (ha=all) | s Subscribe presence | rr (Re-)register |
| H Hold call        | u Unsubscribe presence | ru Unregister    |
| v re-inVite (release hold) | t ToGgle Online status | > Cycle next ac. |
| U send UPDATE      | T Set online status  | < Cycle prev ac. |
| ],[ Select next/prev call | +-----+-----+-----+ |
| x Xfer call        | Media Commands:      | Status & Config: |
| X Xfer with Replaces | cl List ports         | d Dump status    |
| # Send RFC 2833 DTMF | cc Connect port       | dd Dump detailed |
| * Send DTMF with INFO | cd Disconnect port    | dc Dump config   |
| dq Dump curr. call quality | V Adjust audio Volume | f Save config    |
| S Send arbitrary REQUEST | Cp Codec priorities   | f Save config    |
+-----+-----+-----+
| q QUIT             | sleep N: console sleep for N ms |
+-----+-----+-----+
You have 0 active call
>>>>

```

Aperçu des commandes disponibles sous PJSUA

Module interface graphique : TkInter qui est un module de base intégré à Python et que notre équipe est habituée à utiliser.

## 2.2. Composant 2 : Serveur Freeswitch

### *Sprint 1 :*

Serveur : Choix d'utiliser Freeswitch car c'est le serveur de notre projet 1.

### *Sprint 2 :*

Serveur : Freeswitch compatible avec PJSUA. Source : [Freeswitch et PJSUA](#)