

Détermination des composantes

Application de téléphonie IP

1.3

Falimanana Razafindrabe

Johan Yémanlin Sintondji

Joël Villeneuve

29 octobre 2019

# Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2019-11-11 | 1.3 | Première ébauche | SoraTech |

# Définitions

|  |  |
| --- | --- |
| Terme | Définition |
|  |  |

# Abréviations/acronymes

|  |  |
| --- | --- |
| Abré./Acro. | Définition |
|  |  |

# Table des matières

[Historique des révisions 2](#_Toc526337951)

[Définitions 2](#_Toc526337952)

[Abréviations/acronymes 2](#_Toc526337953)

[Table des matières 3](#_Toc526337954)

[1. Introduction 4](#_Toc526337955)

[1.1. Objectifs 4](#_Toc526337956)

[1.2. Portée 4](#_Toc526337957)

[1.3. Références 4](#_Toc526337958)

[2. Description des composants logiciels 5](#_Toc526337959)

[2.1. Composant 1 5](#_Toc526337960)

[2.2. Composant 2 5](#_Toc526337961)

# Introduction

## Objectifs

Le but de ce document est de fournir une description textuelle et graphique sur la structure du logiciel en se basant sur les exigences dans le document des spécifications du logiciel.

## Portée

*[Indiquez si le produit est un système adjacent, un sous-système ou est la mise à jour d’un produit existant. Présentez les systèmes et les logiciels avec lesquels le produit interagit. Aussi, expliquez ce qui ne fait pas partie de la portée.]*

Pour établir la communication de la voix par par IP, Soratech a opté pour l’utilisation d’une librairie existante et de l’adapter au projet actuel. La librairie utilisée est le PJSIP, une bibliothèque de communication multimédia libre et à source ouverte écrite en langage C et mettant en œuvre des protocoles standard tels que SIP, SDP…Le facteur du choix dominant étant sa portabilité. En effet, la librairie peut être importé en python, le langage qui est utilisé pour faire l’interface graphique. Le travail consiste donc a généré le fichier pjsua.pyd de ladite librairie PJSUA. Et ensuite, faire l’interface graphique avec le module Tkinter de Python. Et enfin, établir la communication entre le softphone et le serveur freeswitch. Le projet ne mettra pas en place les fonctionnalités avancées d’un softphone mais pourra faire les fonctionnalités standard tel qu’effectuer un appel, recevoir un appel, envoyer un message, la messagerie vocale…

## Références

*[Énumérez les documents utilisés pour rédiger ce plan. Ex : énoncé des travaux, normes ISO, standards de programmation, …]*

# Description des composants logiciels

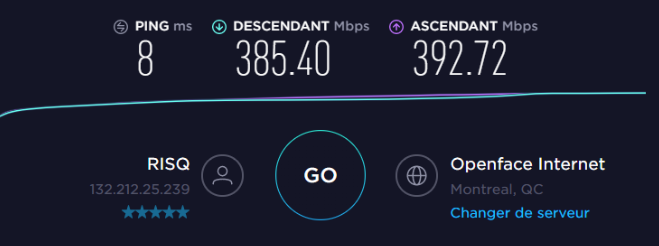
## Ordinateur

Les ordinateurs utilisés pour la phase implémentation du serveur ont les spécifications suivantes :

* Système d’exploitation Windows 10 entreprise 64 bits (10.0, version 16299)
* Processeur Intel® Xeon® CPU E3-1231 v3 @ 3.40 GHz
* Mémoire vive : 8192 Mb
* Cartes réseau : ASIX AX88179 USB 3.0 to Gigabit Ethernet Adapter #2
* Carte réseau : Intel® Ethernet connection I217-LM
* Carte graphique : NVIDIA Quadre K420
* Disque dur : 500 GB

## Connexion

Nous avons testé la connexion internet qui sera utilisé et avons obtenu une bande passante de près de 400 Mb/s descendant et montant.

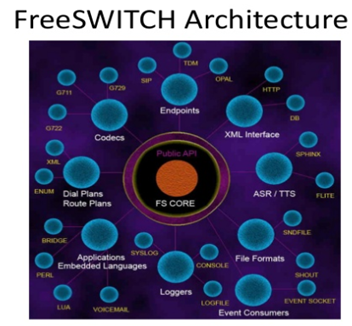


*Figure 1 - Test de la connexion internet*

## FreeSwitch

FreeSWITCH est une bibliothèque qui peut être intégrée à toute application sur n’importe quel appareil allant d’un petit Raspberry Pi qu’a un superordinateur comme les clusters. Cependant, plus généralement, il est conçu pour s'exécuter en tant que service sous Windows. Il utilise CLI comme interface. FreeSwitch est multiplateforme, extensible (ajout facile de fonctionnalités), évolutive, flexible et stable.

FreeSWITCH est construit à l'aide d'un système modulaire. Il a un petit noyau qui fournit le système de base utilisé par tous les modules, mais la plupart des fonctionnalités ne sont pas implémentées dans le noyau lui-même, mais dans des modules autonomes qui ne dépendent pas les uns des autres. Les différents modules communiquent entre eux via le cœur du système.

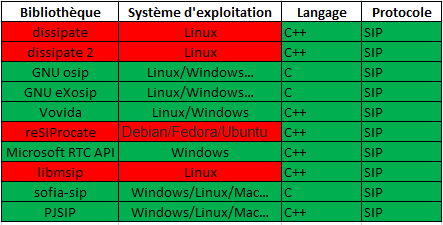


*Figure 2 - Architecture de FreeSwitch*

L'objectif de conception de Freeswitch est de fournir un système modulaire et évolutif autour d'un cœur de commutation stable, ainsi qu'une interface robuste permettant aux développeurs d'ajouter et de contrôler le système.

2-4 Softphone

Plusieurs librairies existent pour établir la communication entre le serveur et le softphone. Les plus populaires sont PJSIP, Osip, Sofia-SIP parmi ceux qui sont développés en c ou en c++. Ces librairies permettent d’établir les fonctionnalités PABX traditionnel pour implanter la voix sur IP. Lors de ce sprint, nous avons établi une liste de ces bibliothèques ainsi que les critères de programmation de chacun. Voici le tableau que nous avons fait:



À la vue de ce tableau, nous concluons qu’il serait possible de faire le projet avec plusieurs librairies. Étant donné que la dernière, PJSIP, est plus documenté et utilisée dans le milieu, c’est celle que nous avons choisi. Nous avons ensuite pu installer PJSUA et il est maintenant prêt et fonctionnel. L’interface graphique du softphone a été développé au cours du sprint 3 et 4. Le rendu visuel est comme suit :  


L’interface est développée en python 3.6.6 avec le module Tkinter. L’implémentation nécessite l’utilisation de la librairie PIL(Pillow) de python pour les images ou les icones utilisés dans le programme. Avec les menus déroulants qui sont créés en cascade, il serait possible de consulter le répertoire, les historiques d’appels et différentes autres options. Le menu a été créés spécialement selon les besoins du clients.