



Architecture et conception du logiciel

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
28/10/19	Version 1	Première version de ce document en ce qui concerne la partie 2 du projet	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. Parties prenantes de la conception et leur préoccupations	5
3. Architecture logicielle	6
3.1. Vue d'ensemble de l'architecture logicielle	6
3.2. Vue d'architecture 1	6
3.2.1. Vue d'ensemble	6
3.2.2. Contraintes de conception qui s'applique à cette vue	6
3.2.3. Exigences et préoccupations de conception	6
3.2.4. Description des éléments de la vue et leurs interfaces	6
3.2.5. Raisonnement	6
3.2.6. Autres vues pertinentes	6
3.3. Vue d'architecture 2	6
4. Conception détaillée	7
4.1. Vue d'ensemble de la conception détaillée	7
4.2. Conception détaillé de l'élément 1	7
4.2.1. Vue structurelle	7
4.2.2. Vue comportementale	7
4.2.3. Autres vues pertinentes	7
4.3. Conception détaillé de l'élément 2	7
4.4. Informations de conception détaillée qui sont pertinentes pour plusieurs vues	7

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

Norme ISO 29110

2. Parties prenantes de la conception et leur préoccupations

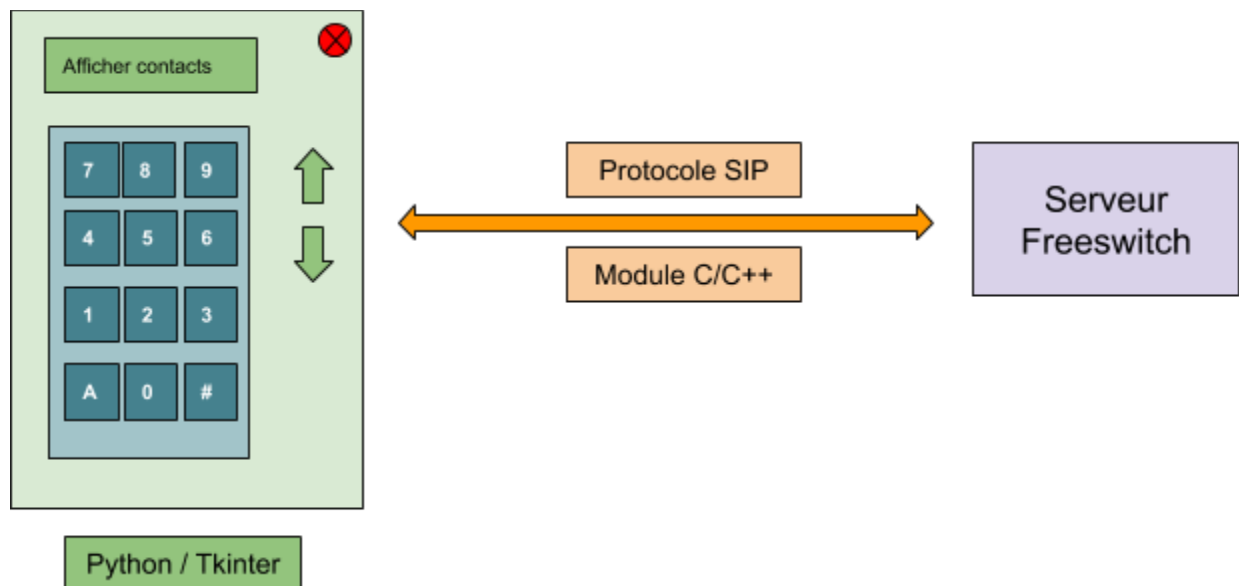
Etant donné que le logiciel sert pour communiquer entre employés ainsi qu'avec des personnes extérieures à l'entreprise, chaque PC sera doté du même logiciel avec les mêmes configurations. En effet, tout le monde disposera des mêmes fonctionnalités. Au moins un technicien aura accès au PC serveur en cas de problème.

De plus, il y a l'équipe de Trifrogz composée de 3 membres avec des rôles de techniciens chargés de l'installation de la solution logicielle terminée chez le client et le Scrum master qui assure l'interface entre les retours clients et son équipe de techniciens. Il est en charge de la bonne conduite du projet.

3. Architecture logicielle

3.1. Vue d'ensemble de l'architecture logicielle

Notre logiciel a pour but de permettre au client d'utiliser le serveur installé dans la première partie du projet. Nous utiliserons une interface graphique simple programmée en langage Python avec le module Tkinter et nous ouvrirons les connections téléphoniques avec un module C++ qui nous permettra de gérer le protocole SIP.



3.2. Vue d'architecture 1 : softphone

3.2.1. Vue d'ensemble

Sur chacun des ordinateurs sera installé notre logiciel permettant la communication à l'intérieur et à l'extérieur de l'entreprise. Celui-ci sera combiné avec le serveur FreeSwitch qui est un logiciel permettant la communication par Internet.

3.2.2. Contraintes de conception qui s'applique à cette vue

Le logiciel doit pouvoir fonctionner sur Windows 10 et ne doit pas surcharger le système, c'est-à-dire que l'employé doit pouvoir se servir d'autres applications en parallèle sans que cela n'affecte le bon fonctionnement de notre logiciel.

3.2.3. Exigences et préoccupations de conception

La communication doit être immédiate et sans délais. Il serait très apprécié que la communication soit aussi sécurisée que possible afin d'éviter toute fuite d'information sensible.

3.2.4. Description des éléments de la vue et leurs interfaces

La vue du logiciel est relativement simple, l'utilisateur pourra ajouter des contacts et avoir accès à cette liste. Il pourra composer n'importe quel numéro de téléphone existant, mettre en attente un appel, le renvoyer à un autre numéro en cas d'absence et mettre en place une boîte vocale personnalisée de l'entreprise.

3.2.5. Raisonnement

La librairie PJSIP est la plus connue de celle permettant d'utiliser le protocole SIP et donc probablement celle qui possède le plus de documentation pour avancer dans le projet. En ce qui concerne Python et Tkinter c'est la méthode la plus simple pour programmer une interface graphique efficacement.

4. Conception détaillée

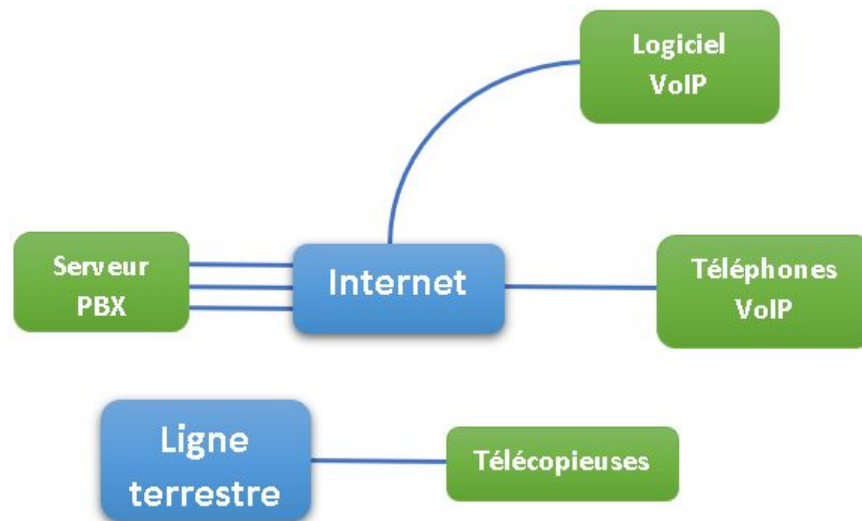
4.1. Vue d'ensemble de la conception détaillée

Comme décrit plus haut, nous avons choisi d'utiliser le logiciel FreeSwitch en tant que serveur PBX afin de constituer le réseau de communication IP de l'entreprise. Pour cela, nous devons tout d'abord vérifier les différentes mises à jour des divers drivers ainsi que de Windows puis nous devons télécharger et installer le logiciel sur chaque ordinateur.

Pour être sûrs que notre logiciel puisse fonctionner, nous devons aussi vérifier que certains logiciels et outils sont déjà installés sur les ordinateurs. Après la configuration du logiciel et la mise en place des périphériques, les employés seront capables de recevoir des appels sur leurs ordinateurs, de les mettre en attente ainsi que de mettre en place une boîte vocale personnalisée ou non.

4.2. Conception détaillée de l'élément 1 : logiciel de softphone

4.2.1. Vue structurelle



4.2.2. Vue comportementale

Notre système permettra d'émettre des appels vers d'autres postes sur un même réseau IP.

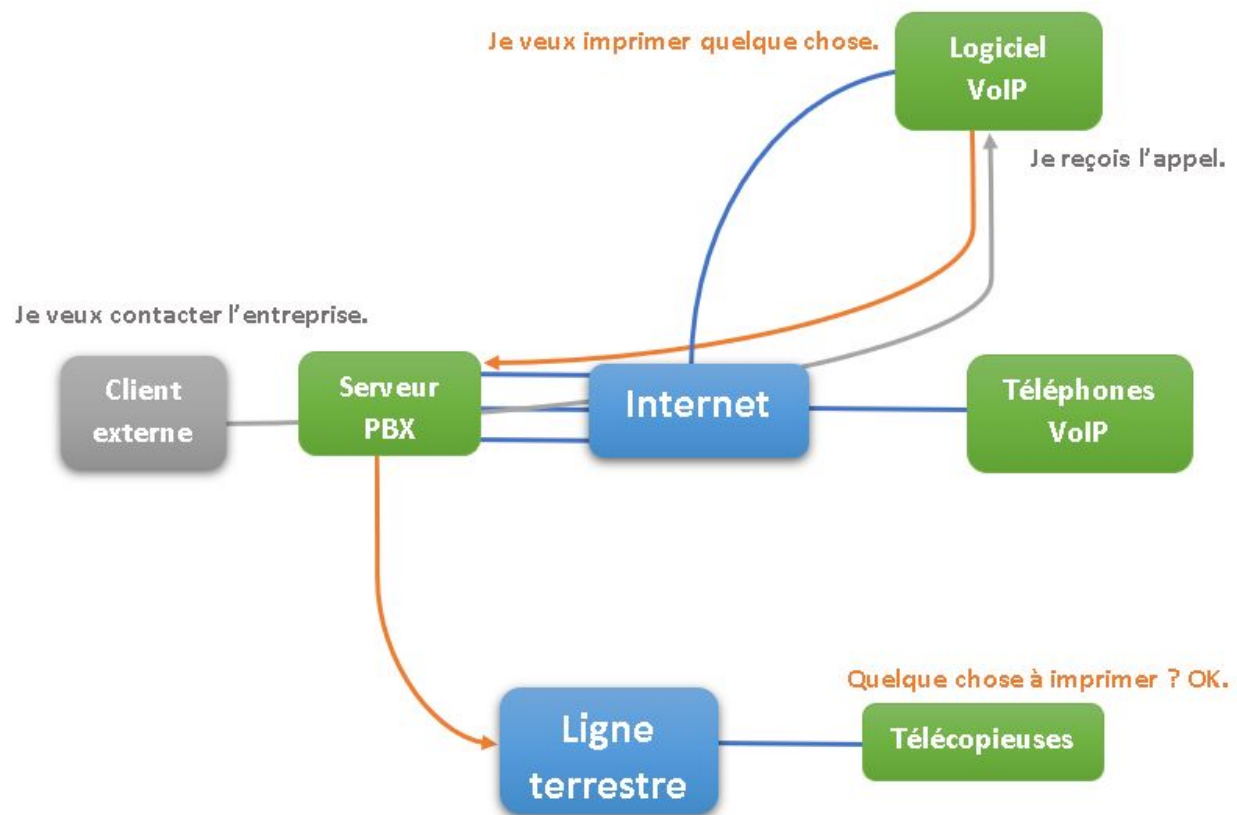
Avec le protocole SIP, un exemple de session d'appels sera basé sur les méthodes suivantes :

Le téléphone appelant envoie une **INVITE**

1. Le téléphone appelé renvoie une réponse informative **100 – Trying – back**
2. Lorsque le téléphone appelé commence à sonner une réponse **180 – Ringing est renvoyée.**
3. Lorsque le correspondant décroche le téléphone, le téléphone appelé envoie une réponse **200 – OK**
4. L'appelant répond par un **ACK** – acknowledgement en anglais
5. Maintenant, la communication est transmise sous forme de données via **RTP**
6. Lorsque l'appelant raccroche, une requête **BYE** est envoyée au téléphone appelant.
7. Le téléphone appelant répond par un **200 – OK**.

[Source : 3cx](#)

Toutes ces méthodes sont gérées par l'API PJSUA implémenté dans notre softphone.



4.2.4. Raisonnement

Le logiciel sera installé sur chaque ordinateur de l'entreprise et devra accomplir les fonctions de base de tout softphone : gérer les appels, renvois d'appel, boîte vocale, appels en attente et gestion

d'une liste de contacts. De plus, le logiciel devra être simple d'utilisation et ergonomique pour que les employés puissent tous l'utiliser aisément.