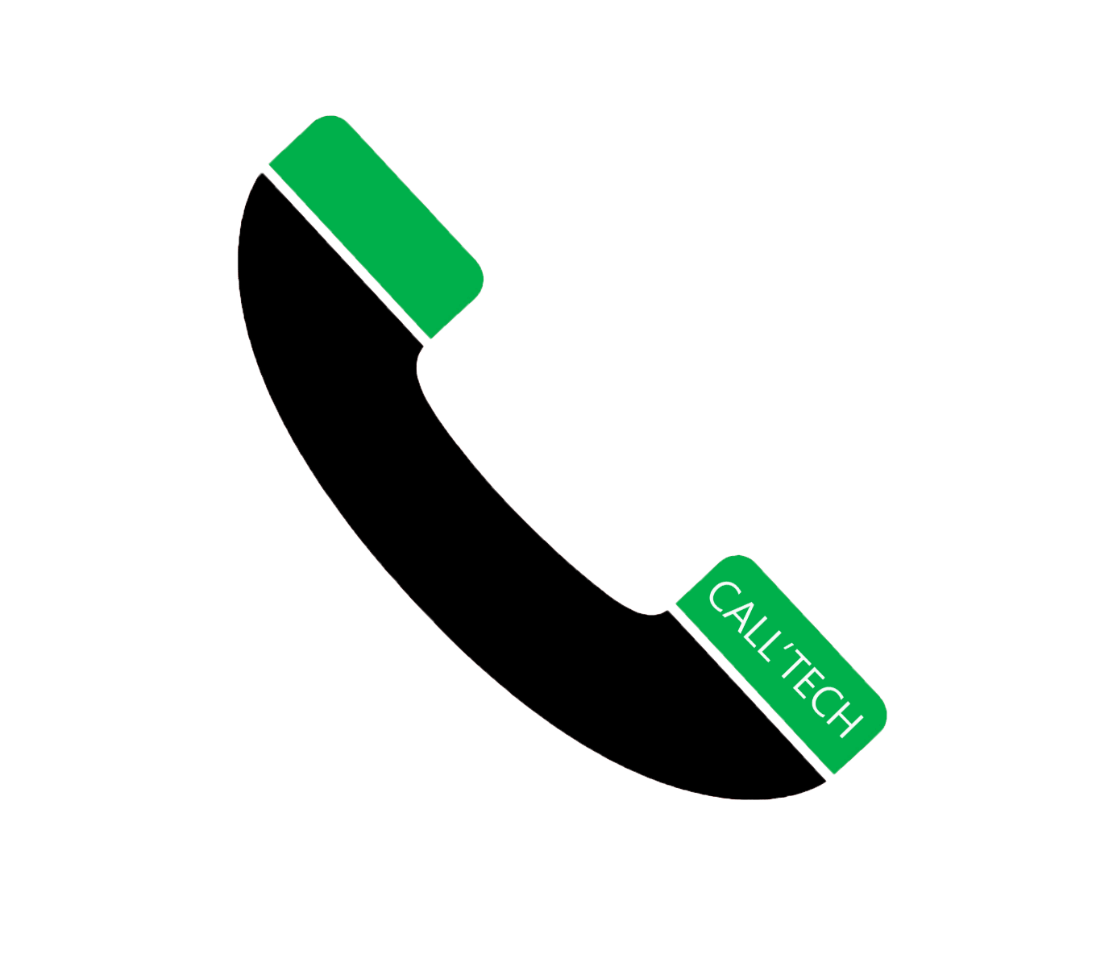
*Semestre 4*

*Projet informatique*



« Call everyone everywhere »

« La téléphonie sur IP, simple, économique et ambitieuse ;   
vous n’allez plus pouvoir vous en passer ! »

Spécifications

**Technologies utilisées : L’équipe :**

- Shell/Perl Alexandre Gerard  
- FreeBSD Bastian Bel-Ange  
- GIT Julie Pradelli  
- SSH, Courrier + Imap, Tinydns  
- Asterisk  
- PHP 7, MySQL, Bootstrap  
  
**Suiveur :** Adjevi KOUDOSSOU

Table des matières

[I. Présentation du projet 3](#_Toc467068274)

[1. Contexte et mission 3](#_Toc467068275)

[2. La plate-forme de services 3](#_Toc467068276)

[A. Présentation des objectifs 3](#_Toc467068277)

[B. Solutions techniques retenues 3](#_Toc467068278)

[II. Glossaire 4](#_Toc467068279)

[III. Services proposés 4](#_Toc467068280)

[1. Objectifs fonctionnels 6](#_Toc467068281)

[A. Préparer le Soekris 6](#_Toc467068282)

[B. Stratégies d’appel différentes pour chaque groupe 7](#_Toc467068283)

[C. Groupes de chasse 7](#_Toc467068284)

[D. Gestion du serveur vocal 7](#_Toc467068285)

[E. Gestion de la conférence 8](#_Toc467068286)

[F. Gestion des appels sortants 8](#_Toc467068287)

[G. Gestion des appels entrants 8](#_Toc467068288)

[H. Gestion des droits d’appel 8](#_Toc467068289)

[I. Développement de l’interface IHM 9](#_Toc467068290)

[2. Objectifs non-fonctionnels 9](#_Toc467068291)

[A. Gestion du fax 9](#_Toc467068292)

[B. Messagerie instantanée 9](#_Toc467068293)

[IV. Architecture cible 10](#_Toc467068294)

[V. Offre de service 10](#_Toc467068295)

# I. Présentation du projet

## 1. Contexte et mission

Dans le cadre de notre semestre 4, nous réalisons ce projet de téléphonie sur IP. Le semestre dernier le projet Beep avait déjà réalisé un projet similaire et avait brillamment réussi.

Nous nous proposons donc de reprendre ce projet en proposant un service de téléphonie sur IP. Il comporte les services fonctionnels suivants : gestion des appels simples, groupés et messagerie et les services non-fonctionnels suivants : gestion du faxe et d’une messagerie instantanée.

## 2. La plate-forme de services

### A. Présentation des objectifs

Chaque utilisateur pourra :

- Appeler des personnes en fonction de son groupe d’appartenance  
 - Personnaliser la réception d’un appel entrant au travers des groupes de chasse  
 - Disposer d’une messagerie vocale et la personnaliser  
 - Organiser une conférence privée (avec mot de passe) ou publique (sans mot de passe)  
 - Accéder à l’interface web pour gérer ses services  
 - Avoir une messagerie instantanée

### B. Solutions techniques retenues

Les solutions techniques retenues sont :

Operating System : FreeBSD 11.0  
Autocommutateur privée : Asterisk  
Serveur web : Nginx  
Serveur mail : Courrier-Imap/Postfix  
Serveur DNS : Tinydns  
Gestionnaire de versions : Git/ SourceTree  
Interface web : PHP7/ MySQL  
Gestion de la connexion au serveur : SSH

Nous disposons de deux machines une machine virtuelle et le Soekris. Ces deux entités sont hébergées sur le réseau itinet.

# II. Glossaire

# III. Services proposés Tous nos objectifs sont liés au serveur de VoIP. Nous commençons donc par un comparatif des différents serveurs VoIP open sources existants pour sélectionner celui qui est le plus adapté à notre projet. Nous détaillerons ensuite chacun de nos objectifs avec le serveur VoIP choisi.

|  |  |  |  |
| --- | --- | --- | --- |
| **Serveur VoIP** | **Asterisk** | **Yate** | **FreeSWITCH** |
| **Logo** | https://upload.wikimedia.org/wikipedia/en/thumb/e/e4/Asterisk_Logo.svg/1280px-Asterisk_Logo.svg.png |  |  |
| **Forces** | **Flexibilité** : le programmeur chevronné peut ajouter des fonctionnalités à Asterisk en Perl, PHP, C et Pascal.  **Documentation abondante** disponible sur le web  **Scalabilté** : on peut partir d’un tout petit système utilisant Asterisk et l’étendre sur plusieurs serveurs  **Richesse des services offerts** : il est très rare de trouver un produit open source avec tant de fonctionnalités | **Flexibilité** : possibilité de création d’applications innovantes grâce à son architecture originale  **Stabilité** : produit plutôt stable  **Scalabilité** : produit adaptable sur différentes tailles de systèmes  **Portabilité** : Yate peut être utilisé dans des applications couvrant plusieurs systèmes d’exploitation et architectures matérielles | **Flexibilité** : Plus de langages supportés qu’Asterisk : .NET, JS, Perl, C++, Python…  **Scalabilité**: encore meilleur qu’Asterisk grâce à une architecture plus adaptée à l’extension  **Documentation abondante** sur le web |
| **Faiblesses** | **Système de configuration complexe** en raison de ses nombreux fichiers de configuration  Gestion administrative par une interface sous Windows peut créer des interruptions plus ou moins graves | **Documentation très pauvre**  Communauté d’utilisateurs et de développeur pour le moment très réduite | **Système de configuration** plus **complexe** que celui d’Asterisk |

Sources : <http://www.blog.saeeed.com/2011/03/logiciels-de-telephonie-ip-vocal-asterisk-yate-comparaison/> et <http://www.ictinnovations.com/asterisk-vs-freeswitch>

En raison de la documentation abondante concernant Asterisk et de sa popularité, nous choisissons ce serveur de VoIP.

## 1. Objectifs fonctionnels

### A. Préparer le Soekris

a. Point de vue utilisateur  
  
Je souhaite pouvoir appeler un correspondant ; c’est le service de base indispensable. Je compose le numéro sur le client de téléphonie (zoiper ou xlite) et je lance l’appel. Au bout de quelques instants, en fonction de la disponibilité de mon correspondant, celui-ci me répond ou on m’envoie sur une messagerie vocale.  
  
b. Point de vue technique

  
Soekris net5501

Pour se faire, nous allons utiliser un Soekris net5501. Ce boîtier placé sur un réseau permet de faire de l’auto commutation téléphonique. Il en existe d’autres tels que le Soekris net6501 qui est plus puissant mais nous choisissons celui-ci car c’est le seul dont nous disposons.  
   
Premièrement, nous allons procéder à la mise en place du serveur PXE sur une machine virtuelle hébergée sur le réseau itinet par les étapes suivantes :  
 - Installer le serveur DHCP, lftp (transfert de fichiers classique) et **tftpd-hpa** (autre transfert de fichier basé sur UDP et adapté pour amorcer un PC sur le réseau)  
 - Compléter le fichier dhcpd.conf pour la configuration de DHCP  
 - Configurer tftp en téléchargeant le pxeboot FreeBSD et un disk de RAM FreeBSD dans le dossier adapté  
 - Configurer le boot.conf de la machine virtuelle  
- Installation de FreeBSD par PXE  
 - Démarrer les services DHCP et tftp  
 - Brancher le câble réseau puis l’alimentation sur le Soekris net5501  
 - Accéder au BIOS du Soekris pour lui indiquer de booter sur le réseau  
 - Se connecter au Soekris par SSH  
 - Vérifier si l’intégralité de l’OS est présent sur le Soekris

B. Stratégies d’appel différentes pour chaque groupe  
  
1) Point de vue utilisateur   
Il y aura plusieurs groupes d’utilisateurs (administrations, enseignants, extérieurs, administrateurs) chacun de ces groupes auront une stratégie d’appel différente   
Cas1 : quelqu’un appelle le service administration, si le service est ouvert (9h-12h30 et 13h30-17h) tous les téléphones du service sonnent et la première personne qui décroche reçoit l’appel et les autres téléphones arrêtent de sonner. Si le service est fermé, l’appel passe sur la messagerie vocale.  
Cas 2 : quelqu’un appelle le service professeur, le téléphone du professeur en question sonne, si le professeur ne répond pas, l’appel passe sur le téléphone portable du professeur, et s’il ne répond pas l’appel passe sur la messagerie vocale.  
Cas 3 : quelqu’un appelle le service administration+ : quelqu’un appelle le service administration+, le téléphone du professeur en question sonne, si le professeur ne répond pas, l’appel passe sur le téléphone portable du professeur, et s’il ne répond pas l’appel passe sur la messagerie vocale.

Il y aura aussi un groupe « occupé » il sera utile si l’un des membres d’un groupe ne veut pas être dérangé et dans ce cas son téléphone ne sonnera pas si il se met dans l’état occupé

2) Point de vue technique

Afin de faire les plans d’appel, il nous suffit de modifier le fichier extensions.conf et configurer les Dialplan.   
Ce sera aussi dans ce fichier que nous allons créer les différents groupes d’utilisateurs

### C. Groupes de chasse

a. Point de vue utilisateur

Je souhaite appeler un membre du service administratif. J’appelle le standard qui me redirige vers leur service. Lorsqu’un utilisateur passe un appel, tous les téléphones des destinataires sonnent :

Cas 1 : Lorsque l’un d’eux décroche, les autres téléphones arrêtent de sonner avec un message : « l’appel a été pris par un membre de l’administration ».

Cas 2 : Lorsque l’option « occupé(e) » est activée, le destinataire ne reçoit pas les appels. Il est cependant obligatoire d’avoir un numéro disponible durant les heures ouvrables.

Cas 3 : Je veux un renseignement spécifique et seuls les téléphones disponibles d’un bureau ou d’une catégorie sonnent.

b. Point de Technique

Encore une fois, nous utiliserons une extension d’Asterisk qui nous permettra d’établir des groupes tels que « administration » afin de zoner les personnes relier à un même corps.

D. Gestion du serveur vocal  
  
a. Point de vue utilisateur

Je souhaite téléphoner à une personne de l’administration. Je téléphone à l’école et je tombe sur un standard. Si je tape 1 cela signifie que je suis un externe. Mon appel ne pourra aboutir que vers l’administration. Si je tape 2, cela signifie que je suis un étudiant. Il me sera alors demandé un mot de passe pour vérifier mon identité. Ensuite je peux choisir de contacter l’administration (le 1) ou un professeur (le 2). Si je choisi un professeur, il me sera alors demandé mon semestre (1 à 10). Si je suis dans la plage d’horaires ouvrables, mon appel aboutira. Si je tape 3, cela signifie que je suis interne à l’école et un mot de passe me sera demandé pour vérifier mon identité. Ensuite, selon une liste de numéros à deux chiffres connus, je peux appeler n’importe qui de l’école.

Sinon, on m’invitera à laisser un message vocal au travers d’une messagerie. Un mail informatif sera envoyé à mon correspondant pour le notifier de mon appel.  
  
b. Point de Technique  
  
Nous utiliserons IVR (Interactive Voice Response) qui permet de guider les utilisateurs vers le service qu’ils souhaitent joindre. On peut également s’en servir pour simplement annoncer les horaires de disponibilité.  
Le standard et les appels viennent tous ici et sont redirigés en fonction des choix de l’utilisateur.  
  
Pour faire parler Asterisk, nous allons installer un moteur de synthèse vocale (Text To Speech). Pour se faire, nous allons mettre en place celui de Google appelé GoogleTTS.

E. Gestion de la conférence  
  
a. Point de vue utilisateur  
  
Je suis un utilisateur et je souhaite organiser une conférence téléphonique. Au travers de l’interface web, deux solutions s’offrent à moi. D’une part, je peux organiser une conférence privée où je sélectionne les participants qui recevront un mot de passe par mail. Seules les personnes possédant ce mot de passe peuvent prendre part à la conférence. D’autre part, je peux choisir de faire une conférence publique ou n’importe qui des personnes internes à IN’TECH peut participer.  
  
b. Point de vue technique  
  
Notre autocommutateur privé est Asterisk. Nous utiliserons donc deux modules d’Asterisk pour permettre cela : DAHDI et MeetMe.  
DAHDI est un composant lié à Asterisk pour gérer la communication entre Asterisk et les différents types de cartes physiques. On installe DAHDI car MeetMe ne fonctionne pas sans DAHDI. MeetMe permet quant à lui de créer des ponts de conférence. Pour se faire, nous suivrons les étapes suivantes :  
  
- Installer DAHDI sur le Soekris  
- Se mettre dans le même répertoire qu’Asterisk, télécharger et installer DAHDI  
- Démarrer DAHDI  
- Installer MeetMe (toujours sur le Soekris)  
- Arrêter Asterisk  
- Faire apparaître le menu select de MeetMe et y sélectionner les bons modules  
- Recompiler Asterisk avec MeetMe  
- Relancer Asterisk et vérifier que l’application est bien présente à l’aide de la console Asterisk

### F. Gestion des appels sortants

### G. Gestion des appels entrants

### 

### H. Gestion des droits d’appel a. Point de vue utilisateur Lors de mon inscription sur l’interface IHM, j’ai choisi le groupe auquel j’appartiens. Si je suis quelqu’un de l’administration, je suis joignable sur les horaires ouvrables par les personnes extérieures et intérieures à l’école sauf si je décide de me mettre en mode « occupé(e) » pour ne pas être dérangé(e). J’ai bien évidement le droit de téléphoner à quiconque faisant partie des groupes internes à IN’TECH. Cependant, je suis restreint pour téléphoner vers l’extérieur. Tous les appels métropolitains sont permis mais les appels à l’étranger sont interdits. b. Point de vue technique En fonction du contexte auquel appartient l’utilisateur.

Dans le sip.conf d’Asterisk, lors de la création des comptes, on précise à quel contexte appartient chaque utilisateur. Ensuite, dans le extension.conf, on attribut certains dialplans restrictifs pour certain groupes et d’autres dialplans pour plus permissifs pour d’autres.

I. Développement de l’interface IHM  
  
a. Point de vue utilisateur

Je me rends sur le site calltech.itinet.fr. Une fois mon compte crée, je télécharge le client Zoiper et je n’ai plus qu’à suivre les instructions sur le site. Je dois renseigner les informations nécessaires comme le nom, le prénom, le numéro de téléphone et le groupe auquel je souhaite être assigné. Mon assignation au groupe choisi sera en attente de validation jusqu’à ce qu’un administrateur valide celui-ci. Une fois le client installé, je configure mon compte afin de pouvoir être joignable à tout moment et de pouvoir joindre d’autres personnes.

b. Point de vue technique

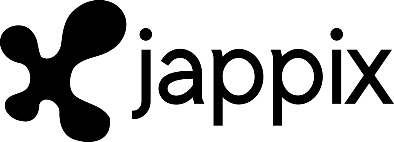
Pour héberger notre site, nous utiliserons un serveur avec Nginx. Pour ce qui est du site en lui-même, nous utiliserons le modèle MVC avec PHP7. L’interface visible sera une page web dans le style one-page. Le client sera opérationnel (identifiants de connexion valides) une fois l’inscription finalisée.

## 2. Objectifs non-fonctionnels

### A. Gestion du fax

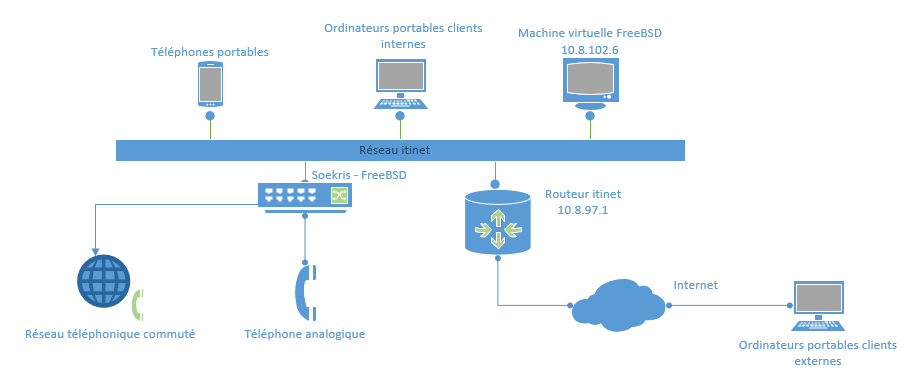
### B. Messagerie instantanée

a. Point de vue utilisateur  
  
Je suis un administrateur interne à IN’TECH et je veux communiquer avec mon collègue trois bureaux plus loin pour une précision. Je me connecte à mon compte via Jappix et j’accède à mon gestionnaire de messages. Je clique sur new conversation et je sélectionne mon destinataire. Je peux ainsi dialoguer de manière efficace et simple avec mes collègues.

b. Point de vue technique  
  
  

Pour cela, nous utiliserons un serveur XMPP ; OpenFire et un client XMPP : Jappix pour la communication entre les utilisateurs.

# IV. Architecture cible



# 

# V. Offre de service

