

---

## I. RPC : question de cours

**Question 1 :** Quels sont les caractéristiques des **API RPC** *haut* et *bas* niveau ? Avantages et inconvénients ? Donner des exemples d'applications pour lesquels vous choisiriez plutôt l'une que l'autre ?

---

## II. RPC : Serveur de Mailing-lists

L'objectif est d'étudier l'implémentation d'un service RPC de *listes de diffusions* de courriers électroniques, ou *Mailing lists* (ML). Un serveur de ML maintient des listes identifiées par un nom (*chaîne de caractère*, par exemple "i2" ou "i4info"), chaque liste regroupant les adresses email des utilisateurs abonnés à cette liste (une liste chaînée de listes chaînées). Chaque utilisateur abonné à une ML peut envoyer une requête de diffusion d'un fichier au serveur de ML qui va alors diffuser ce fichier par email à tous les abonnés de la liste. Les fichiers diffusés seront vus comme des tableaux d'octets de taille variables. On suppose que l'on dispose d'une fonction `int Send-Mail(char * EmailFrom, char * EmailTo, char * Subject, int Size, unsigned char *Data);` qui permet d'envoyer par email le tableau de données `Data` de taille `Size` au destinataire `EmailTo` depuis l'émetteur `EmailFrom` avec le sujet `Subject`.

Le serveur RPC qui gère les ML doit offrir les services suivants :

**ML\_Create()** : création d'une nouvelle mailing list. Cette fonctionnalité ne doit être accessible qu'à un administrateur identifié par un mot de passe (*chaîne de caractères*) passé en 1er paramètre de la fonction. Le 2ème paramètre est le nom de la liste. Le résultat est 0 si rejeté ou 1 si accepté.

**ML\_Delete()** : destruction d'une liste de diffusion. Fonction duale de la précédente.

**ML\_Subscribe()** : demande d'abonnement d'un utilisateur qui fournit *son adresse email* et le *nom de la liste* à laquelle il souhaite s'abonner. On supposera que n'importe qui peut envoyer une requête d'abonnement ou de désabonnement. Le résultat est 0 si rejeté ou 1 si accepté.

**ML\_UnSubscribe()** : demande de désabonnement d'un utilisateur. Fonction duale de la précédente.

**ML\_Send()** : envoi *un fichier* pour diffusion, *l'email de l'émetteur* et le *nom de la liste* sur laquelle il faut le diffuser. Le résultat est 0 si rejeté ou 1 si accepté.

**ML\_List()** : renvoie **une liste chaînée** des noms des ML gérées par le serveur de ML (Attention : pas les listes des emails de chaque mailing lists, uniquement les noms des mailing lists)

**Question 1 :** Faites un schéma de la structure de donnée **au niveau du serveur** pour maintenir les informations nécessaires à la gestion des listes de diffusions. Donner les déclarations en C de ces structures.

**Question 2 :** Donnez les **déclarations RPCL** des **structures de données** dont vous allez avoir besoin **pour les services RPC** en expliquant le rôle de chacune.

**Question 3 :** Donnez la **spécifications RPCL** complète du serveur (inutile de répéter les déclarations de la question précédente).

**Question 4 :** Donnez le code C de la fonction `ML_List()` du **serveur**. Attention de gérer l'allocation de la mémoire de façon à éviter toute fuite.