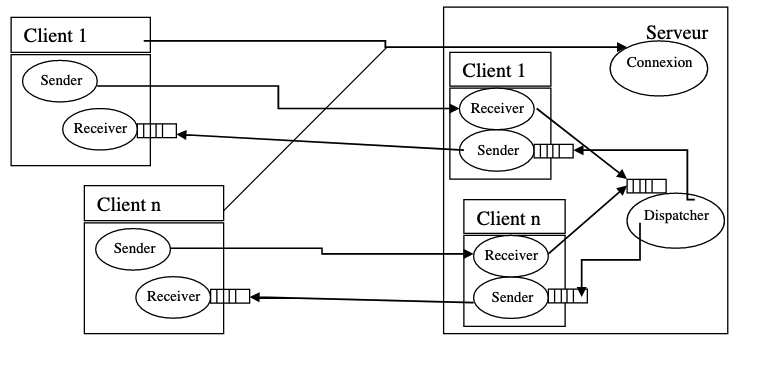
Réalisation d’une application multitâche de type “ client / serveur ”

**Jean Mopin & Victor Bultez**

Notre projet se décompose en deux : une partie client et une partie serveur.

Le programme serveur possède un “ socket ” associé à un numéro de port spécifique. Il est en attente d’une demande de connexion.

Le client effectue une demande de connexion. Il connaît la machine hôte et le numéro de port auquel est connecté le serveur.

Le serveur accepte la connexion, il attribut un nouveau socket et un nouveau numéro de port. De son côté, le client attribut également un socket et un numéro de port. A ce moment, 2 flots de données sont établis entre les deux machines. Le client et le serveur peuvent communiquer en écrivant ou en lisant dans leur socket.

SOCKETS TCP

Les fichiers : *Serveur.java & LaunchServer.java* permettent respectivement d’initialiser le Socket via *ServeurMessage* et le ServeurSocket via la class *LauchServeur* avec le numéro de port.

ECHANGES ENTRE LE CLIENT ET LE SERVEUR

Pour échanger des données, le serveur et le client utilisent les flux d’entrée/sortie fournis par leur socket : la fonction *getInputStream()* de socket pour la lecture et la fonction *getOutputStream()* pour l’écriture.

*Flux d’entrée :*

Le principe est simple, on va obtenir un flux d’entrée avec la fonction *getInputStream(),* créer un nouveau flux pour convertir les octets en caractères (char), créer encore un flux avec tampon pour lire ligne par ligne pour enfin pouvoir lire la chaîne de caractères.

*Flux de sortie :*

Le principe est fondamentalement le même. Il est cependant nécessaire de préciser que nous forçons le vidage du buffer au travers de la fonction *flush().*