

# PolyMessages — Sprint 3

Eri Agnese, Marvin Bontemps, Lucas Nougier

15 Mai 2022

## Contents

|          |                                   |          |
|----------|-----------------------------------|----------|
| <b>1</b> | <b>Protocole de communication</b> | <b>2</b> |
| <b>2</b> | <b>Architecture</b>               | <b>4</b> |
| <b>3</b> | <b>Répartition du travail</b>     | <b>4</b> |
| <b>4</b> | <b>Exécution du code</b>          | <b>5</b> |
| <b>5</b> | <b>Difficultés</b>                | <b>5</b> |

# 1 Protocole de communication

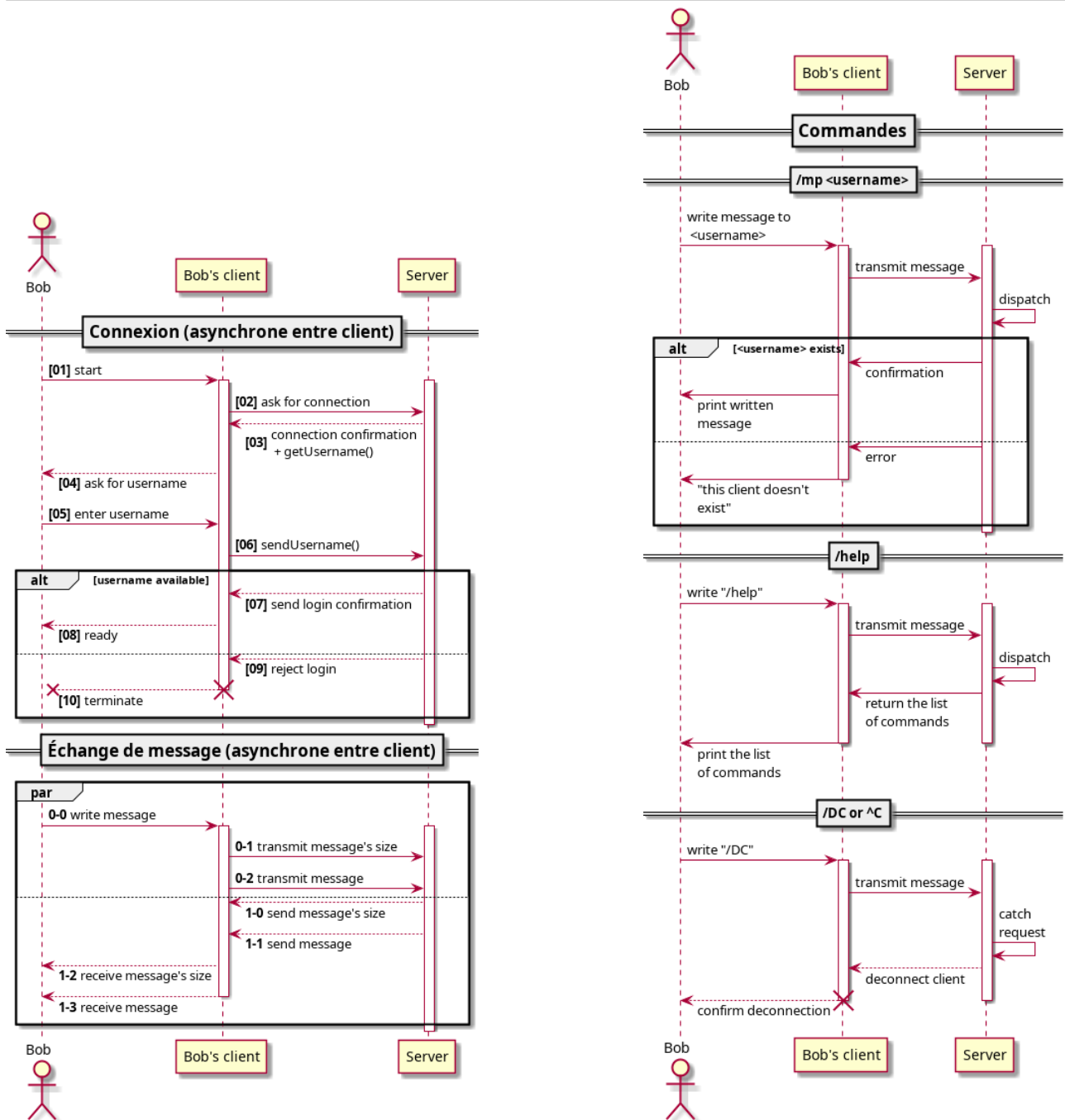


Figure 1: Protocole de communication clients/serveur

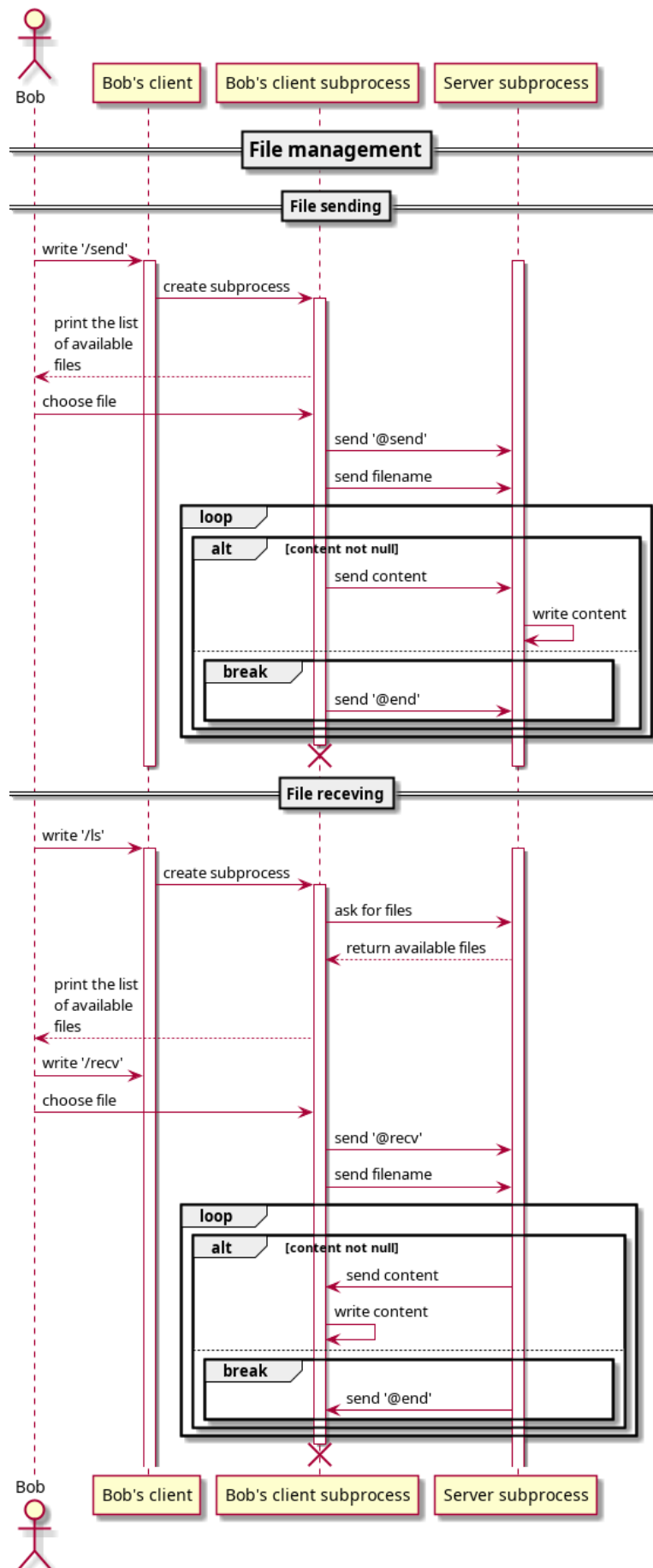
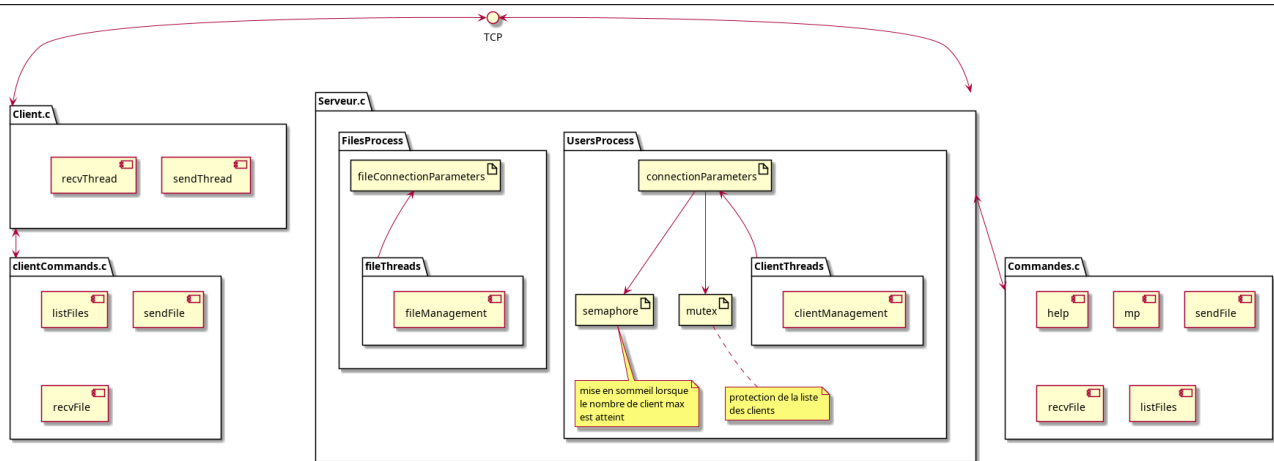


Figure 2: Protocole d'échange de fichiers

## 2 Architecture



**Figure 3:** Architecture de la messagerie

## 3 Répartition du travail

La 3<sup>ème</sup> version de la messagerie consistait à donner la possibilité d'envoyer des fichiers depuis un répertoire précis. Pour cela, on a permis à l'utilisateur de lister les fichiers disponibles sur le serveur et sur sa machine. Puis on a implementé des fonctions d'envois et de réception.

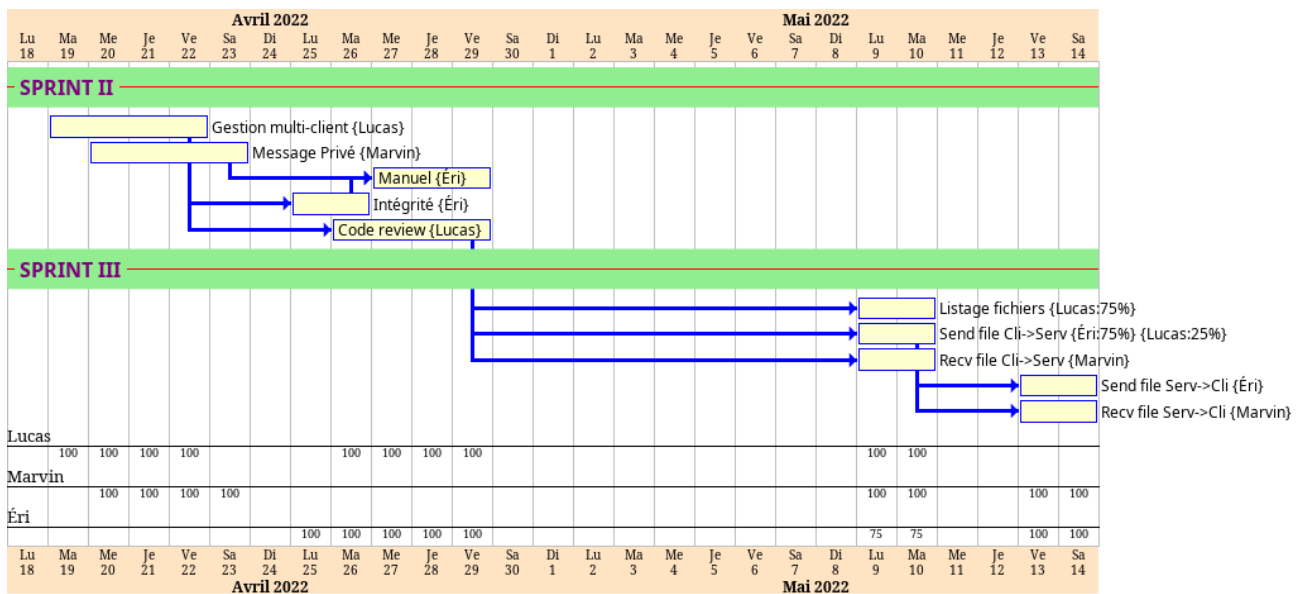
Pour éviter que les envois soit bloquants (pour le client et pour le serveur), on a choisi de gérer les fichiers :

- Sur le client via un processus lourd qui sera lancé à chaque envoi/réception et arrêté lorsque sa tâche sera accomplie
- Sur le serveur via un processus lourd (qui lance un serveur écoutant sur un 2nd port) et pour chaque client qui fera une requête sur des fichiers, un thread lui sera associé.

Pour l'instant, il n'y a pas de gestion des fichiers de même noms.

| Tâches                   | Étudiants |     |        |
|--------------------------|-----------|-----|--------|
|                          | Lucas     | Éri | Marvin |
| Listage fichiers         | X         |     |        |
| Envoie client→serveur    | X         | X   |        |
| Reception client→serveur |           |     | X      |
| Envoie serveur→client    |           | X   | X      |
| Reception serveur→client |           | X   | X      |
| Intégration              | X         |     |        |

**Table 1:** Tâche effectuée par étudiant



*Figure 4: Diagramme de Gantt sur la réalisation du projet*

## 4 Exécution du code

Pour lancer la messagerie, il faut commencer par compiler et lancer le serveur en spécifiant les 2 ports (1: port d'écoute pour les messages, 2: port d'écoute pour les fichiers)

```
[lucas@xps-lucas ~]$ gcc -o server serveur.c commandes.c
[lucas@xps-lucas ~]$ ./server <PORT1> <PORT2>
```

On peut alors lancer les clients en spécifiant les mêmes ports que pour le serveur

```
[lucas@xps-lucas ~]$ gcc -o client client.c clientCommands.c
[lucas@xps-lucas ~]$ ./client <IP> <PORT1> <PORT2>
```

## 5 Difficultés

- On ne peut envoyer qu'un seul fichier par client, après, cela ne fonctionne plus