

UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 585 – Télématicque

Travail pratique #2

Travail présenté à
M. Bessam ABDULRAZAK

par

Raphaël Cayouette (15 064 430)
Jean-Philippe Croteau (15 072 957)
Jonathan Martineau (15 059 124)
Félix Vigneault (15 079 576)

Le 7 juillet 2017

La description sommaire du système-solution

Le serveur est responsable de contrôler l'accès et la modification aux fichiers. Il utilisera des connexions TCP sur des sockets afin de recevoir les commandes des clients. Les clients feront eux-mêmes des demandes de synchronisation et seront responsables de fournir les informations nécessaires au serveur pour qu'il retourne toutes les nouvelles modifications.

La description des modules principaux

Serveur

Afin de préserver la cohérence des données, le serveur devra utiliser un processus de mise en attente d'opérations afin de pouvoir recevoir plusieurs commandes de clients différents, en même temps, sur des connexions TCP différentes. Il devra aussi gérer le fait que les clients ne seront pas nécessairement à jour dans leur version locale des fichiers lors de leurs demandes d'opérations. Des procédures seront mises en place pour gérer les cas de multiples opérations sur un même fichier par plusieurs clients différents.

Certaines opérations auront une importance plus élevée et effaceront les opérations d'importance inférieure lorsqu'elles arrivent de façon simultanée. Par exemple, si un client demande la suppression d'un fichier et qu'un autre demande la modification de ce même fichier en même temps, une des deux opérations sera traitée en premier par le serveur, mais peu importe l'ordre des opérations, c'est la suppression qui va l'emporter puisqu'elle a une importance plus élevée que la modification.

Ordre d'importance des opérations :

1. Suppression
2. Modification avec la date de modification la plus récente
3. Ajout avec la date de modification la plus récente

Le serveur aura aussi comme rôle de gérer les droits d'accès aux informations concernant les utilisateurs, les groupes et les dossiers. Il devra donc garder les informations concernant les clients, les groupes, la composition des groupes, la structure des dossiers (fichiers présents) et le contenu des fichiers. Cette information devra être persistante afin de ne pas être perdue lors d'un redémarrage ou une panne du serveur.

Le contrôle d'accès se basera sur un système d'authentification où les clients fournissent un nom d'utilisateur et un mot de passe afin de s'identifier. Le serveur gardera une trace de qui est connecté grâce à un timestamp de dernière opération. Les demandes des clients devront être validées lors de chaque appel afin de s'assurer qu'ils sont connectés et que leur demande est valide selon leurs appartenances aux groupes et leur statut dans le groupe (administrateur ou membre).

Client

Le client aura une version locale des dossiers auxquels il a accès qui sera placé dans un dossier racine de son choix. Il sera responsable de lui-même analyser les modifications qu'il fait sur les fichiers. Pour ce faire, il sera en mesure de calculer un résumé de l'état courant de ses fichiers en itérant sur tous les fichiers de ses dossiers pour construire une liste contenant toutes les informations sur les fichiers. En conservant le résumé de l'état de la dernière capture, le client pourra refaire une capture et comparer ces deux captures pour déterminer les fichiers ajoutés, modifiés et supprimés.

Le client est responsable de communiquer correctement ses modifications et son état local au serveur afin que le serveur le mette à jour correctement. Il doit aussi tenir le serveur informé de son état connecté/déconnecté lors des connexions et déconnexions.

Communication et connexion TCP

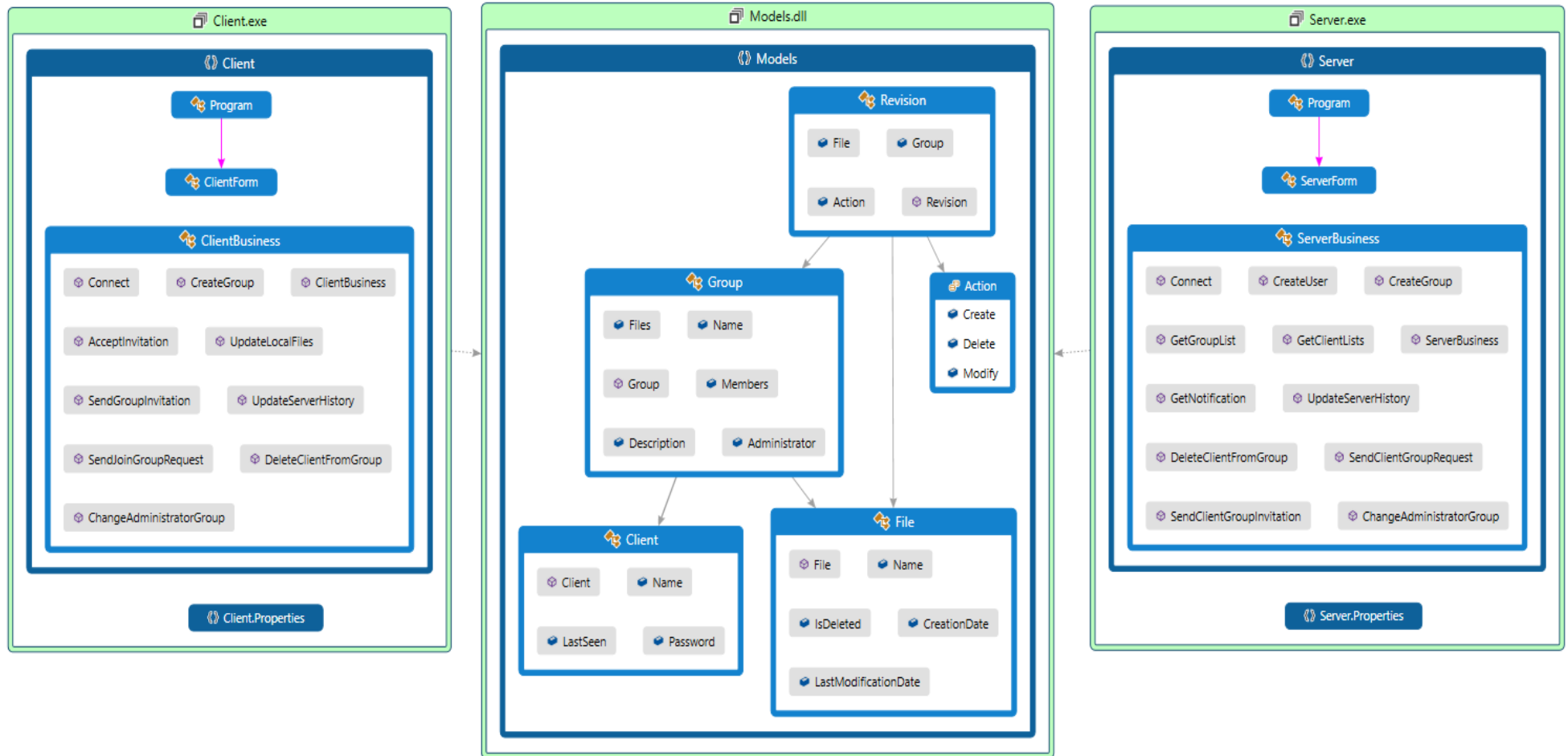
Pour synchroniser les opérations locales, le client calculera ses modifications locales ainsi que le résumé de l'état courant pour les envoyer au serveur, qui choisira quelles modifications peuvent être appliquées. Le client transférera ses fichiers en même temps que ses demandes d'ajout et de modification pour que le serveur puisse appliquer directement les modifications à sa version interne si elles sont valides. Dans la réponse du serveur, le client pourra retrouver les dernières modifications faites par les autres clients et pourra alors mettre à jour sa version locale des fichiers. De cette façon, le client sera complètement synchronisé lors de chaque vérification récurrente (à toutes les 15 secondes).

Pour que le serveur puisse communiquer avec un client, le client devra créer une connexion TCP avec le serveur pour envoyer ses commandes. Pour que le serveur puisse analyser correctement les commandes, il doit connaître la longueur des messages envoyés par le client (et vice-versa). Pour ce faire, nous allons préfixer chaque message avec la longueur du message à recevoir. Donc, chaque message comprendra 32 bits au début qui indiquent la taille de ce qu'il reste à recevoir, ce qui permet d'envoyer jusqu'à 4 Gb dans un seul message. Dans un message, il y aura un entête de taille fixe qui permettra de connaître le type de message pour aller récupérer son contenu correctement.

Format d'un message:

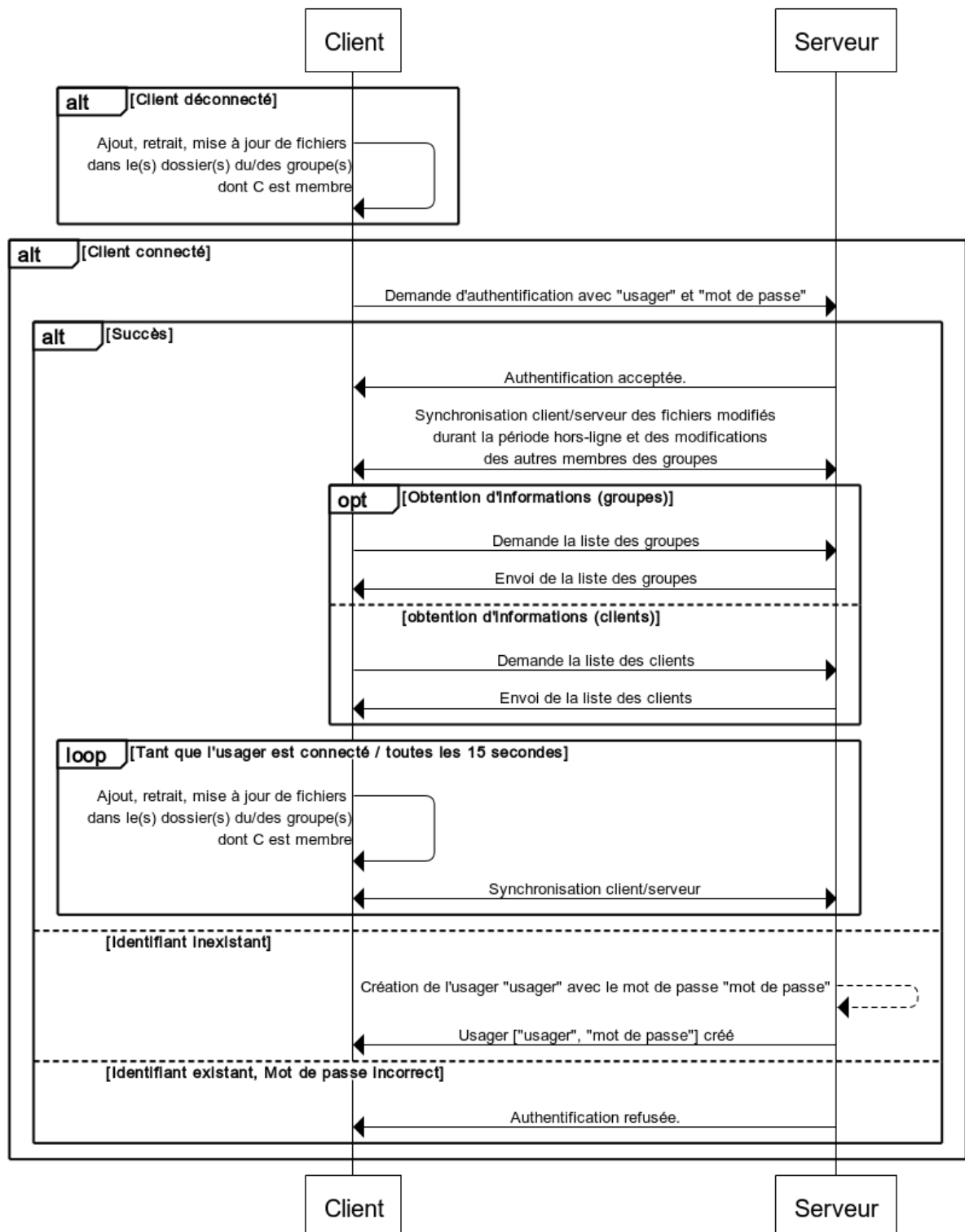
Taille du message (32 bits)	Type du message (16 bits)
Nom d'utilisateur (255 o)	
Contenu du message	

Le diagramme de classes

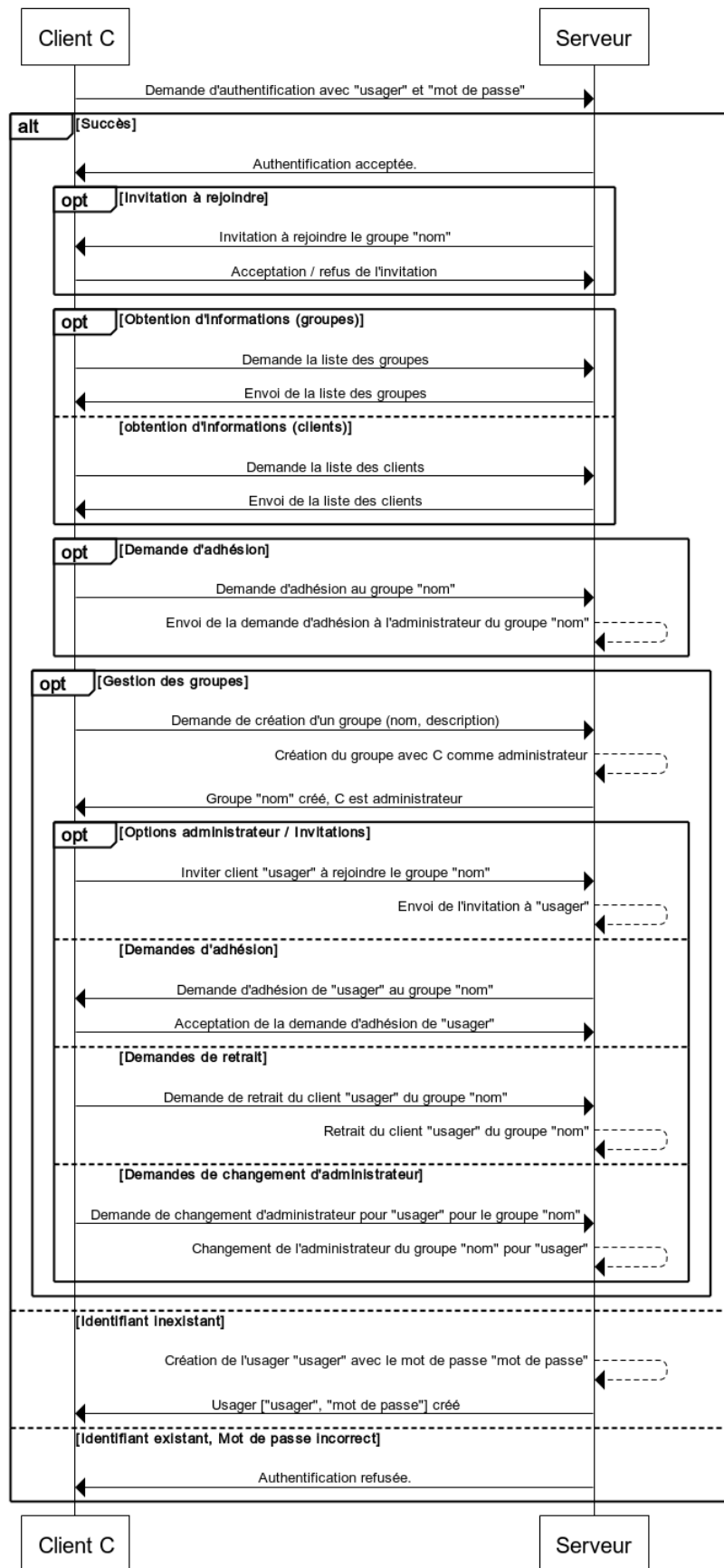


Les diagrammes de séquence

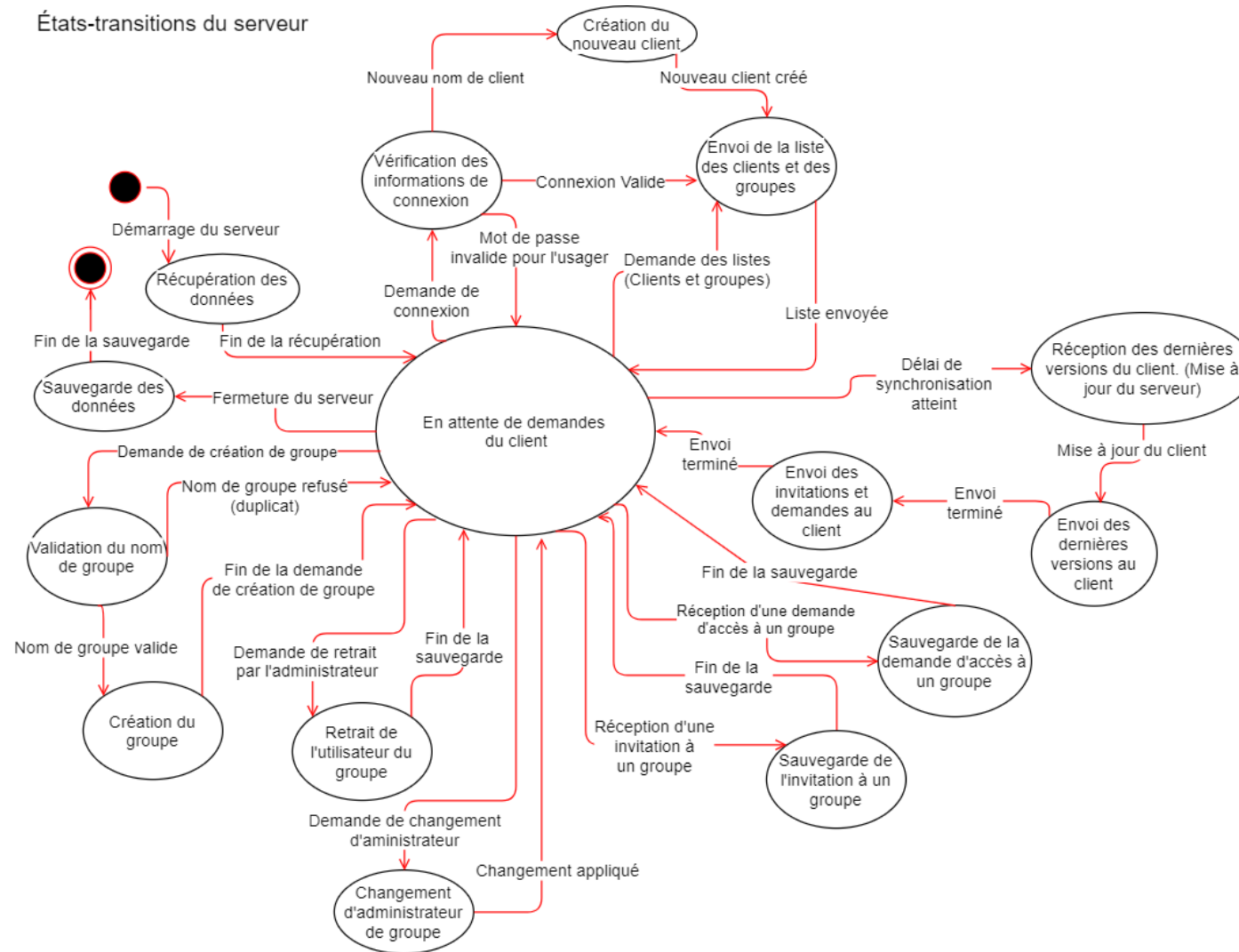
Séquence de gestion des fichiers



Séquence de gestion des groupes



Les diagrammes d'états-transitions



États-transitions du client

