Kould Not Konect

Document de conception générale

Projet **Kould Not Share** v1.0

Version	Date	Description
V.1	16/11/14	Première version de la conception générale

Kevin BASCOL, Kevin LAOUSSING, Nicolas REYNAUD

Sommaire

1 Introduction

1.1 Portée du document

Le logiciel de client/serveur FTP est un outil permettant à des particuliers ou des entreprises l'échange de donnée de manière contrôlée. Par exemple, chaque utilisateur propriétaire d'un fichier stocké dans le serveur FTP du logiciel, pourra paramétrer les droits de téléchargements sur ce fichier, et ainsi il disposera du pouvoir de restreindre ou d'augmenter l'accessibilité de son fichier...

1.2 Objectif du document

Ce document présente la conception logicielles et matérielles de la version 1.0 du projet Kould Not Share de l'entreprise Kould Not Konnect. Les responsables de ce projet sont Nicolas Reynaud, Kevin Laoussing et Kevin Bascol.

1.3 Définitions, acronymes et abréviations

KNK Kould Not Konect.

KNS Kould Not Share.

FTP File Transfer Protocol, protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP.

MVC : Modèle-Vue-Controleur, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Protocole Spécification de plusieurs règles pour un type de communication particulier.

Client Logiciel qui envoie des demandes à un serveur.

Serveur Dispositif informatique matériel ou logiciel qui offre des services, à différents clients.

Downloader Anglicisme du mot "télécharger".

Uploader Anglicisme du mot "téléverser".

2 Description générale

Ce projet consiste en la création d'un serveur FTP chez un particulier ou une entreprise, ayant accès à la machine sur laquelle est installé le programme. Les utilisateurs se verront donner l'accès par l'administrateur à certains dossiers de la machine sur laquelle est installé le serveur. Ils pourront faire alors des échanges de fichiers dans ces répertoires à l'aide du client FTP.

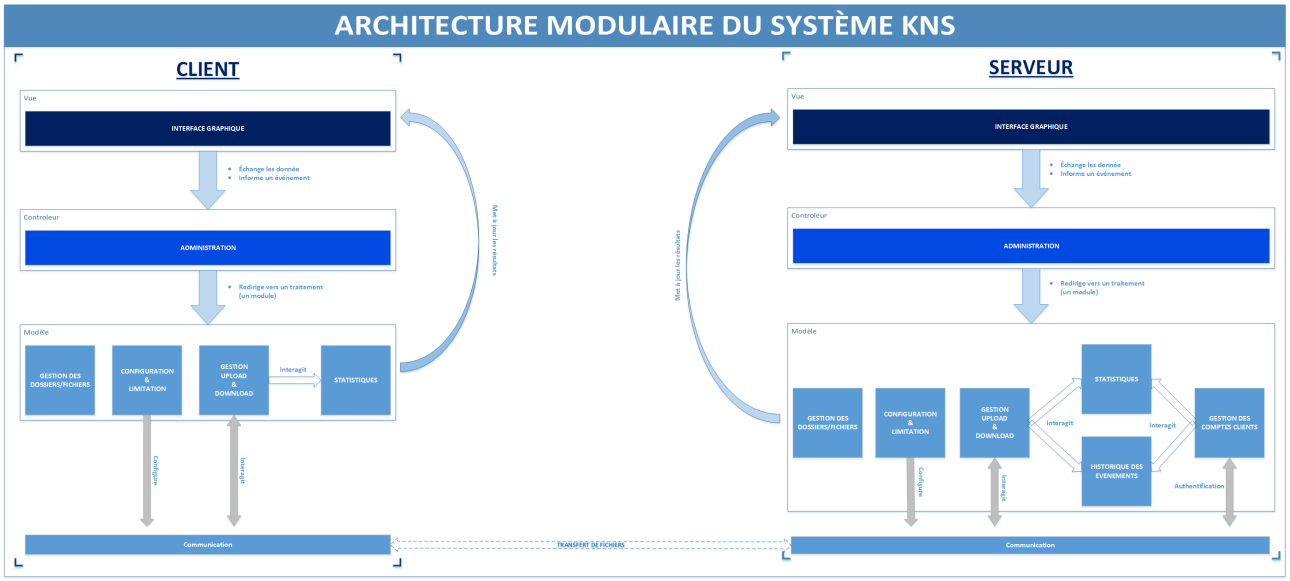
3 Architecture du système Kould Not Share

Le logiciel Kould Not Share est composé en deux sous-systèmes : le client et le serveur. Cette partie décrit la conception d'un point de vue modulaire et modèle, du logiciel KNS.

3.1 Architecture modulaire du système

3.1.1 Schéma de la décomposition modulaire du système

Ce schéma présente d'une part le patron utilisé par le système et d'autre part la décomposition modulaire du système.
Il décrit de manière général le fonctionnement du système à travers les interaction entre les modules.



(cf Annexe.)

3.1.2 Description des modules

3.1.2.1 Modules Client

3.1.2.1.1 Module interface graphique

Le module de l'interface graphique contiendra toutes les classes Java implémentant l'interface graphique du client KNS. Le module utilisera principalement l'API Swing de la bibliothèque JFC. Il doit accomplir les tâches suivantes :

- Présenter les informations,les données et les résultats renvoyés par un traitement.
- Recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.).
- Effectuer correctement le traitement lié à l'événement.

Ce module ne doit effectue pas de traitement, il doit se contenter d'afficher les résultats des traitements demandés par l'utilisateur.

3.1.2.1.2 Module d'administration

Le module d'administration est l'ensemble des classes qui "administrent" et coordonnent les traitements selon les actions de l'utilisateur. Il doit accomplir les tâches suivante:

- Prendre en charge la gestion des événements de synchronisation pour mettre à jour l'interface graphique selon le traitement effectué.
- Recevoir tous les événements de l'interface graphique et enclencher les traitements à effectuer, autrement dit, enclencher le traitement d'un module.

Ce module ne doit effectuer aucun traitement sur la requête de l'utilisateur. Il doit analyser l'action de l'utilisateur et se contenter d'appeler le traitement du module adéquat.

3.1.2.1.3 Module de gestion des dossiers

Le module de gestion des dossiers regroupe toutes les classes qui implémentent la gestion des dossiers. Ce module doit accomplir les tâches suivantes :

- les traitements de manipulation de fichiers et de répertoires (cf Document des spécifications des exigences, section 3.1.1.4).
- les traitements d'affichages (cf Document des spécifications des exigences, section 3.1.1.4).

Par conséquent le module doit interagir avec le système de l'utilisateur (manipulation de fichiers et de répertoires), et le module de communication (pour obtenir l'arborescence d'un répertoire héberger dans un serveur FTP).
Le module ne doit effectuer de traitement que lorsqu'il est appelé par le module d'administration.

3.1.2.1.4 Module de configuration et limitation

Le module de configuration et limitation regroupe toutes les classes qui implémentent les configurations du client et les limitations des transferts de fichiers du module de communication (cf Document des spécifications des exigences, section 3.1.1.3).
Le module doit interagir avec le module de communication pour lui appliquer la configuration décidée par l'utilisateur.
Le module ne doit effectuer de traitement que lorsqu'il est appelé par le module d'administration.

3.1.2.1.5 Module de upload/download

Le module de upload/download regroupe toutes les classes qui implémentent les opérations d'upload et de download d'un fichier (cf Document des spécifications des exigences, section 3.1.1.6).
Le module doit interagir avec le module de communication et le systeme du client pour effectuer un transfert de fichiers(upload et download).
Le module doit interagir avec le module de statistiques pour comptabiliser les opérations effectuées par le module de upload/download.
Le module doit respecter la configuration donnée par le module de configuration et de limitation. Le module ne doit effectuer de traitement que lorsqu'il est appelé par le module d'administration.

3.1.2.1.6 Module de statistiques

Le module de statistiques regroupe toutes les classes qui implémentent toutes les opérations de statistiques décrites dans le documents de spécification des exigences (cf Document des spécifications des exigences, section 3.1.1.7).
Le module doit effectuer des traitements à chaque opérations de transfert de donnée du module de upload/download, et lorsqu'il est appelé par le module d'administration.

3.1.2.1.7 Module de communication

Le module de communication regroupe toutes les classes qui implémentent la communication de réseau. Le principale traitement de ce module est effectuer tout les types d'opérations réseau du logiciel. Ce module doit accomplir les taches suivantes :

- Permettre au client de se connecter à n'importe quel serveur FTP.
- Effectuer toutes les opérations d'authentification (cf Document des spécifications des exigences, section 3.1.1.5).
- Effectuer toutes les opérations de transfert de fichier en respectant le protocle FTP et la configuration imposée par le module de configuration et de limitation (cf Document des spécifications des exigences, section 3.1.1.6).

Le module doit interagir avec le module de upload/download lors d'un transfert de fichiers.

3.1.2.2 Modules Serveur Le serveur reprend les mêmes modules que le client, mais s'ajoutent des modules supplémentaires.

3.1.2.2.1 Module des historiques des événements

Le module des historiques des événements regroupe les classes qui implémentent la gestion d'historique des événements et le stockage de l'historique. Il doit donc accomplir les tâches suivantes :

- Insérer dans un fichier les événements qui sont survenus (création de compte, suppression de compte etc)
- Stocker les actions survenues dans un fichier texte.
- Afficher les événements survenus.
- Éditer un rapport une fois par jour contenant les événements de la journée.

3.1.2.2.2 Module de gestion des comptes clients

Le module de gestion des comptes clients regroupe les classes qui implémentent le stockage et les droits des comptes. Elle réalise donc les actions suivantes :

- Stocker les comptes dans différents fichiers textes.
- Vérifier si un compte existe.
- Vérifier si le mot de passe est correct.
- Crée un compte.
- Supprimer un compte.
- Bannir un compte.

3.2 Modèle MVC

Notre logiciel utilise un modèle MVC pour la partie client et la partie serveur.

3.2.1 Rappel

3.2.1.1 Principe de MVC

Le modèle MVC permet de bien séparer les données, la présentation et les traitements correspondant respectivement aux trois parties suivantes : le modèle, la vue et le contrôleur (voir section 1.1 Schéma du système).
Pour rappel, lorsqu'un utilisateur interagit avec l'interface graphique de l'application (qui correspond à la vue du patron MVC), celle-ci envoie une requête :

- la requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données) ;
- le contrôleur demande au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée ;
- la vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement "via" le modèle).

3.2.1.2 La vue

3.2.1.2.1 Description

Elle fournit une présentation du modèle. En général, la vue reçoit directement du modèle l'état et les données qu'elle doit afficher.

3.2.1.2.2 Modules composant la vue de MVC

- Module de l'interface graphique.

3.2.1.3 Le contrôleur

3.2.1.3.1 Description

Il accepte les entrées des utilisateurs et détermine ce qu'elles signifient pour le modèle.

3.2.1.3.2 Modules composant le contrôleur de MVC

- Module administration.

3.2.1.4 Le modèle

3.2.1.4.1 Description

Le modèle contient les états, les données et la logique applicative. Il n'a pas conscience de la vue ni du contrôleur, même s'il fournit une interface pour accéder à son état et le manipuler et s'il peut informer les observateurs des changements d'état.

3.2.1.4.2 Modules composant le modèle de MVC

Pour la partie client :

- Module de gestion de dossiers/fichiers;
- Module de configuration et limitation;
- Module de gestion de upload/download;
- Module de statistiques;

Pour la partie serveur :

- Module de gestion de dossiers/fichiers;
- Module de configuration et limitation;
- Module de gestion de upload/download;
- Module de statistiques;
- Module de gestion des comptes clients;
- Module de gestion des historiques des événements.

3.2.2 Conception du modèle MVC

Le modèle MVC est un patron de conception composé : il intègre plusieurs patrons de conceptions décrite dans cette section.

3.2.2.1 Observateur

Le modèle implémente le patron de conception Observateur, afin que les objets intéressés soient mis à jour quand un changement d'état se produit. L'emploi du patron de conception Observateur garantit que le modèle demeure complètement indépendant des vues et des contrôleurs. Il permet également d'utiliser des vues différentes avec le même modèle, voir d'en utiliser plusieurs en même temps.

3.2.2.2 Stratégie

La vue et le contrôleur mettent en œuvre le patron de conception Stratégie : la vue est un objet qui est configuré avec une stratégie. Le contrôleur fournit la stratégie. La vue n'est concernée que par les aspects visuels de l'application et délègue au contrôleur toutes les décisions sur le comportement de l'interface. L'emploi du pattern Stratégie permet également de découpler la vue du modèle puisque le contrôleur a la responsabilité d'interagir avec le modèle pour exécuter les requêtes des utilisateurs. La vue ne sait absolument pas comment il procède.

3.3 Interaction entre module

3.3.0.3 Client cf. Annexe, Diagramme de séquence client.

3.3.0.4 Serveur cf. Annexe, Diagramme de séquence serveur.

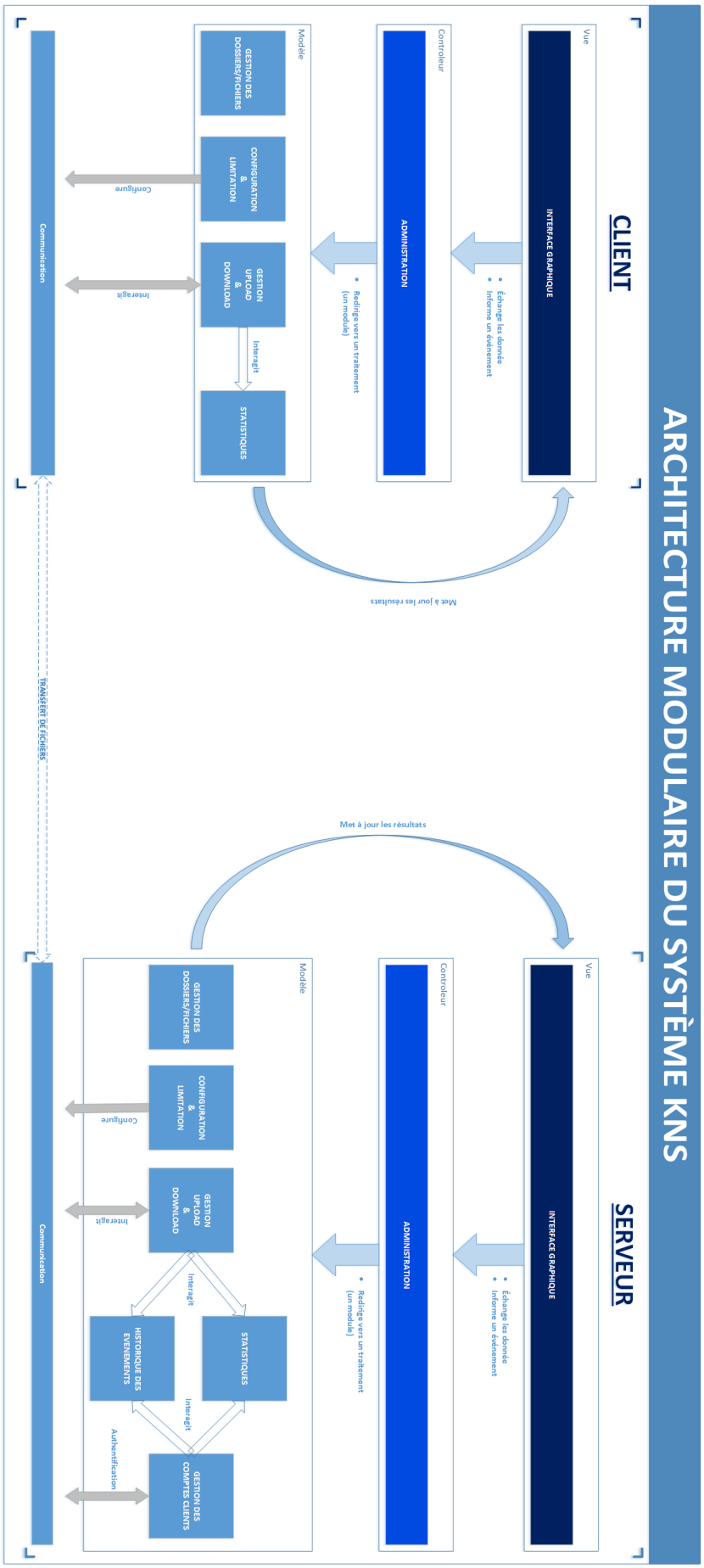
4 Définition des principales structures de données

4.1 Structures de données

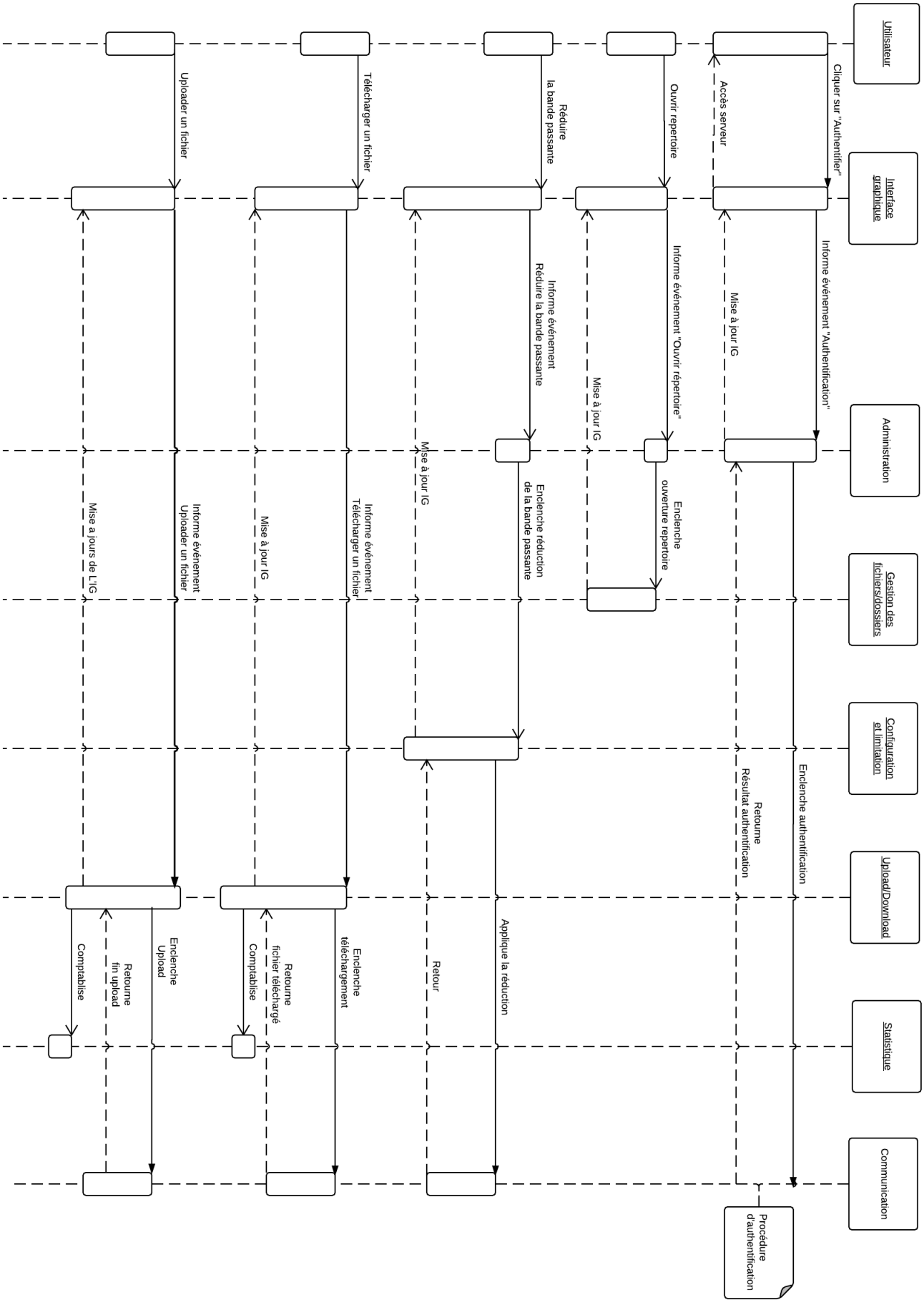
4.2 Interactions entre les structures de données

5 Annexe

5.1 Architecture modulaire du système KNS



5.2 Diagramme de séquence client



5.3 Diagramme de séquence client