The client’s purpose is to allow users an easy way to download and share files through the p2p application.

Features

* Shared files search – found files can be either locally or remotely resident.
* File share – choose a file in the computer and start sharing it.
* File download – choose a remote file and start downloading it.
* Stop sharing a file.

GUI

Yet do be decided. All the actions of the program can be exposed through CLI or GUI.

Files Manager

The files manager is a single object through which the client can query for available files, download files, share and more.

The file manager holds 3 important members – the file sharing server, the server communication channel and a list of file transfers.

Synchronization with the server

All files’ actions must start by contacting the server. This is for us to have the most relevant information before we start performing actions.

If for example we would’ve cached information of shared files from previous sessions we might reach faulty states in which once we’ll change the server the application would think files A, B and C are shared, although the new server won’t be notified of this information. For this reason, we must first contact the server and synchronize with it.

Communication channel

A single object which governs the communication with the server.

The communication channel should receive messages to send to the server and retrieve responses from it.

File sharing

File sharing is done via a local sharing server, running in a different thread.

The file sharing will be done by accepting new clients, and opening a new FileTransfer thread for each one.

The logic of the file sharing server is similar to the server himself and might be shared between them.

File transfer

A file transfer represents a single p2p line of communication between 2 clients.

In the file transfer the current client can either be the server (in case of a file sharing scenario) or the client (in case of a file download scenario).

In either case – the file transfer runs in a separated thread and continues transferring data until it is either done or broken.

It exposes an API that allows its holder to query its state, stop it, and get arbitrary data from it (such as the result code for example).

The file transfer itself will perform the network actions using its own communication channel.