# Architecture of the system

There are three types of nodes; the client node, the name node, and the data node. The client node can give either the APPEND command or READ command to the name node, which acts as both a server to the client node but a client to the data node.

The APPEND command takes in the name of the file to append and the content that the user wants to append to that file of that name as arguments. The file to be appended is searched for using the name node and is appended by creating a new data block and then sending the block to be stored in a data node. If the file does not exist, then an error message is returned by the APPEND command. No message is returned is the writing to the file was successful.

The READ command takes in a file name as an argument. The READ command uses the name node to return blocks of data whose contents when ordered, make the contents of the original file. In other words, the data blocks are concatenated in such a way that when combined, become exactly like the original file.

# Why our design decisions

We utilize threads to ensure that our name node can handle multiple requests. Each request is processed as a thread. Each of these threads is given to a name node handler. The name node handler connects to every data node. After each data node is connected to, the data nodes create a data node handler which is responsible for processing each of the requests that were given as threads.

# Description of the tests conducted on our system