## CECS 327 Lab Project: P2P

**Group member:** Guanyu Ding, Riana Jara, Andrew Chheang

When talking about problems we encountered while making the project, at first, we did not consider whole folders (containing files) also being synced when a node is in the network. We solved this problem by implementing the Directory class with two HashMaps: one for storing file data and one for storing sub-directory data. By using this Directory in our File Controller class, we can then keep track of folders being synced as well. Another problem we encountered was in terms of nodes updating files on the local needing to be updated on the fellow nodes within the network. We fixed this issue by having the File Controller register the update as a Send File Event. Lastly, one minor issue we had was when the node leaves the network, we weren't sure whether or not the files should stay synced or not. We decided to go with the route of having files and sub-directories in the local node send remove events to the nodes in the network, so that once the leaving node has left the network, all copies of its files/folders are also deleted off the neighboring nodes.

Some explanation of the code (also found on the ReadMe on our GitHub: <https://github.com/rianajara/CECS327_TermAssignment1>)

## Event

This synchronization network is an event based network. All operations are triggered by different events.

**Event Types:**

|  |  |
| --- | --- |
| **Event Name** | **Description** |
| Join Event | For joining the network, new node will broadcast this event |
| Join Response Event | After receiving a join event, the node will send back this event |
| Leave Event | Telling all other nodes that a node is leaving |
| Remove File Event | Telling all other nodes that a certain file is removed |
| Send File Event | Send a file to other nodes |

## 

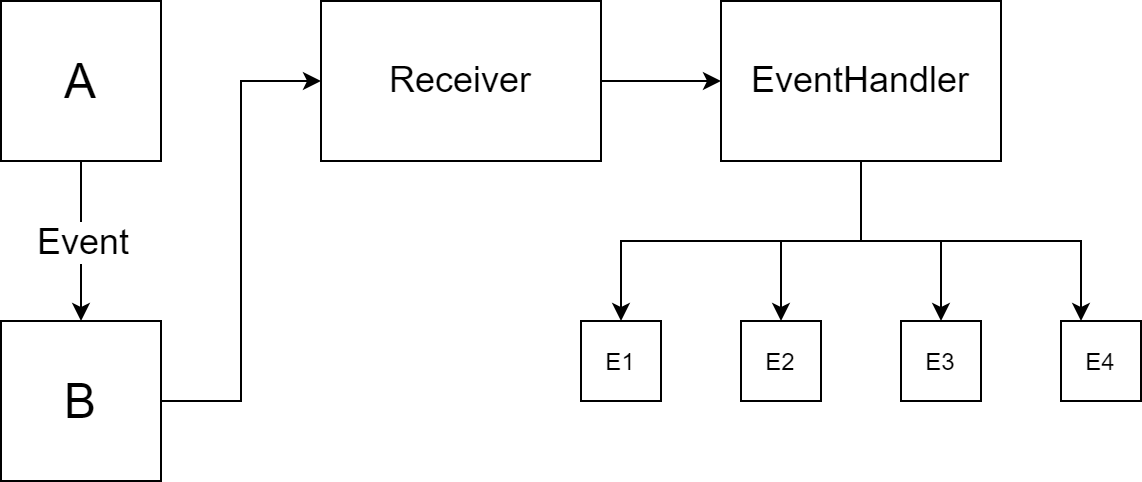
## File Controller

File controller is one of the core components of this entire program. It will scan the local files information again and again to make sure the file record in the program is the latest. If it finds the modification of files, it will send the related event to other nodes. For example, if a directory is removed locally, then it will tell other nodes to remove this directory as well.

## Event Handler

Event handler is another core component of this program. It will be created in Node entity and be passed into sender. When a node receives event data from other nodes, the event handler will resolve the data and determine what kind of event arrives. Then it will do different kinds of operations based on the type of the event.

## Example



For example, Node A is going to send a file to Node B. Node A will create a sendFileEvent entity which contains some key information for sending a file such as the path, filename, data of the file, etc. And Node A will send the entire entity to Node B. The receiver of Node B will receive the event data and pass it to the event handler. The event handler will resolve the data and determine the type of the operation. Then, it will take out the data and create the corresponding files on Node B.