**SecureCEdit: P2P Secure Document Editing Framework**

**Phase 1:**

1. Setting up P2P subnetworks

* Two files Controller and Peer
* Controller:

1. Manages the connection between all the peers
2. All the peers are connected to this Controller using socket programming and multi-threading with Controllers socket address and port number
3. Peers register to the Controller by entering their names and connect to them with their respective port numbers
4. Creates sub networks between peers using Hash maps and Serialization
5. Broadcasts messages between all connected peers
6. Peer-to-Peer messaging between the peers of the sub networks

2. Setting up broadcast messages

* Peers connect to the Controller by registering themselves
* Controller sends broadcast messages to all the peers connected
* A new message sent whenever a peer joins a network
* A new message sent whenever a peer disconnects from a network

3. Prototype an editor

* A text editor built using Java Swing
* It includes a text area, menu bar with 3 options: File, Edit and Close
* File with options: New, Open and Save
* Edit with options: Cut, Copy and Paste
* Close
* Save:

1. Creating and downloading the credentials file from <https://console.developers.google.com/apis/> by creating a new API Console project, enabling the Drive API and generating the Client ID and Client Secret
2. Google Drive API to read the user credentials file and creating a service
3. Connecting user to user’s Google account prompting the user for user credentials to login to the account
4. Saving the file on the user’s Google Drive
5. Providing the File ID of the saved file