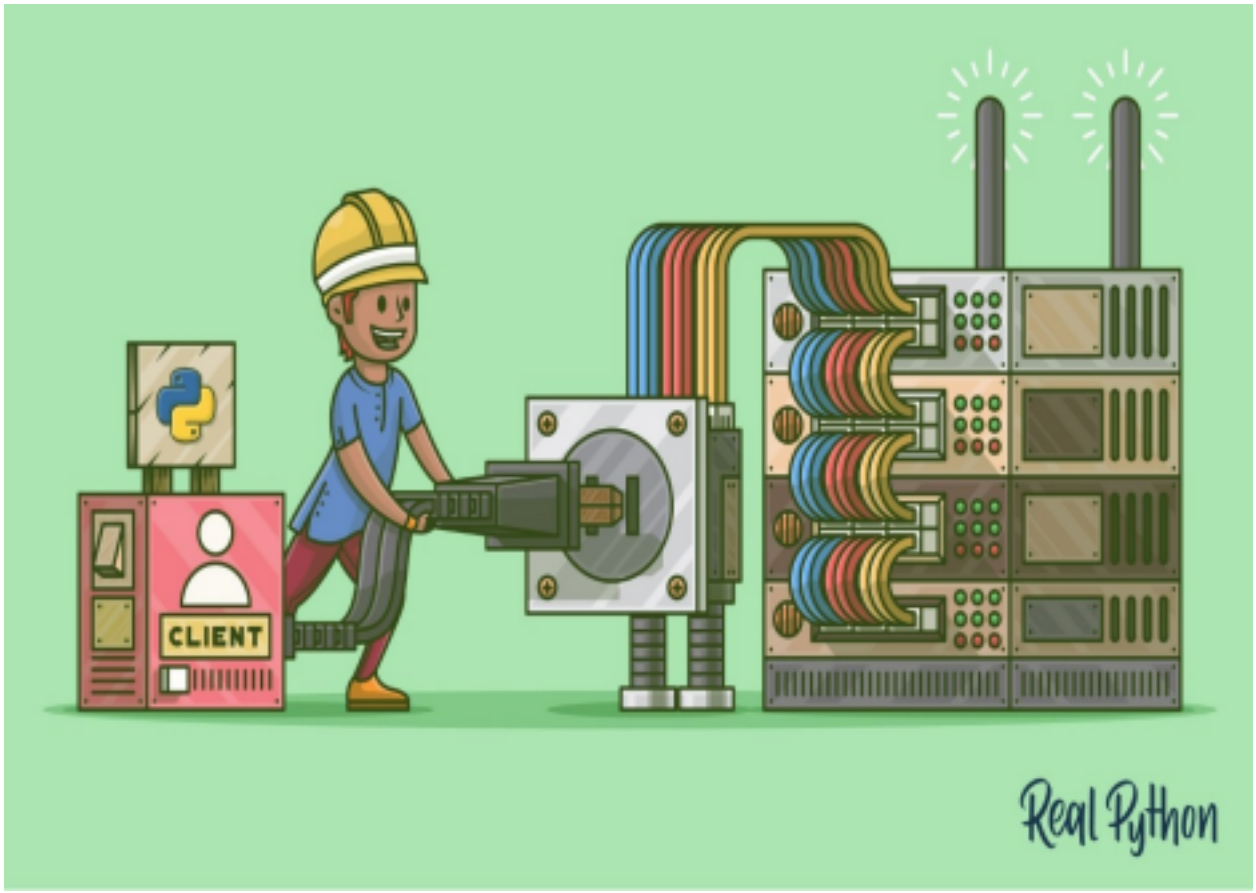
JEC Industrial Project



TEAM 17

TEAM NAME: The 4’s Squad

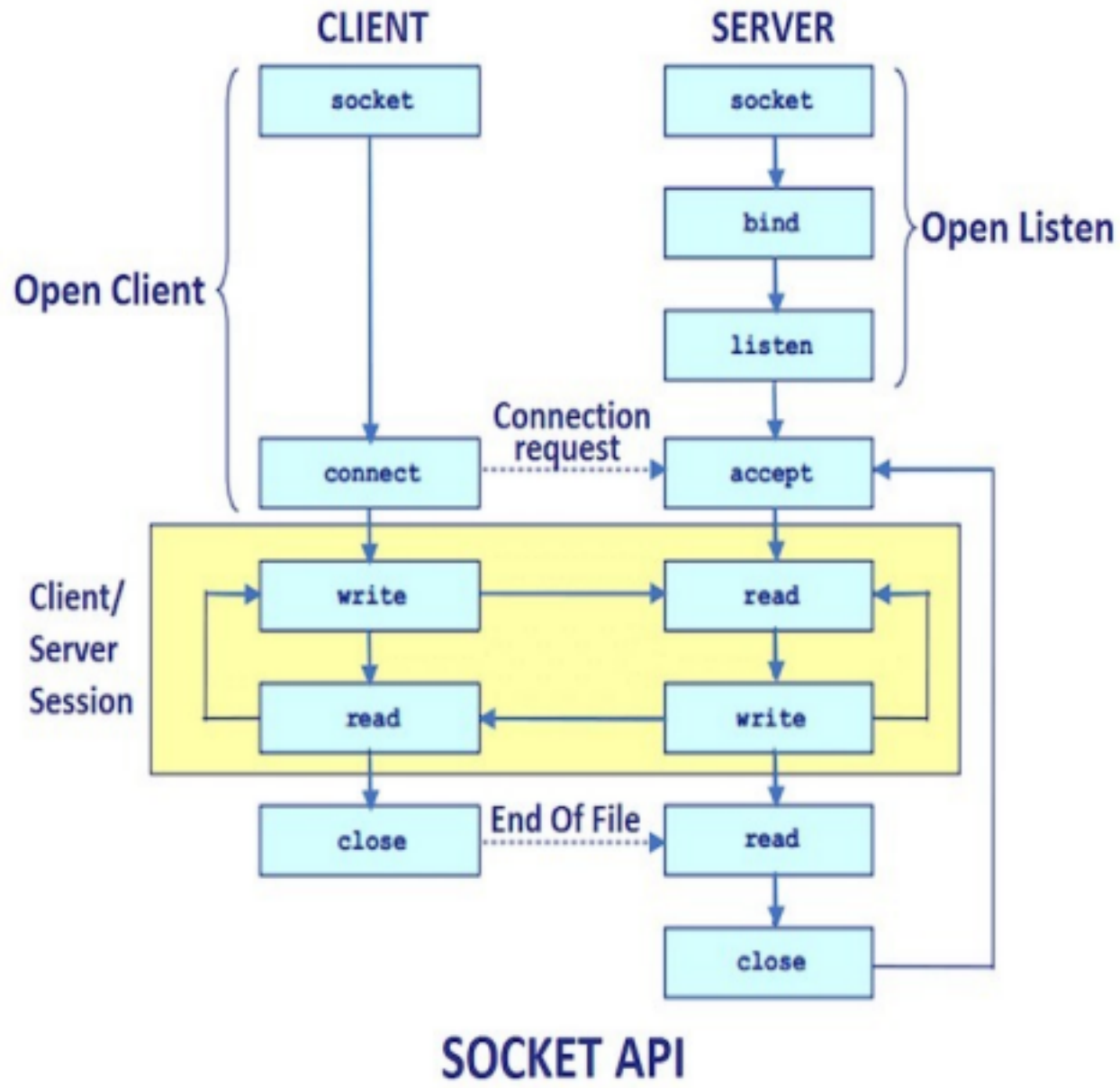
TEAM MEMBERS

- Akshat Bansal
- Prabhanshu Raj Jain
- Srishti Kulchandani
- Shristi Vishwakarma

Goal

This Project aims to build a group based **file sharing system** (like Bit Torrent) where users can share , download files from the group they belong to. Download should be parallel with multiple pieces from multiple peers.

Server-Client Program



1. Socket Creation:

Socket is a data communication endpoint for exchanging data over the network.

- Uniquely identified by:
  - ip address
  - end-to-end protocol (e.g. TCP or UDP)
  - port number

Command line:

```
int servSocket = socket(AF_INET, SOCK_STREAM, 0);
```

2. Inet_pton:

The Inet_Pton function converts an IPv4 or IPv6 Internet network address in its standard text presentation form into its numeric binary form. The ANSI version of this function is inet_pton.

Command line:

```
inet_pton(AF_INET,"0.0.0.0",&servAddr.sin_addr);
```

3. Bind:

After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure).

Command line:

```
//Binding the server with the port
   if (bind(servSocket, (sockaddr *)&servAddr, sizeof(servAddr)) < 0)
   {
      serverFile << "Error in Binding \n";
      cout << "Error in Binding \n";
      return -1;
   }
```


4. Listen:

It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog defines the maximum length to which the queue of pending connections for sockfd may grow. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

Command line:

```
//Listening
   listen(servSocket, SOMAXCONN);
```

```
serverFile << "Started listening to : " << Port << endl;
cout << "Started listening to : " << Port << endl;
```

5. Accept:

The server gets a socket for an incoming client connection by calling accept().

Command line:

```
//Accepting
int dataRecv = recv(clientSocket, data, 2048, 0);
```

For Client

1. Socket connection:

Exactly the same as that of server's socket creation.

2. Connect:

Command line:

```
int connect(servSocket, (sockaddr *)&servAddr, sizeof(servAddr));
```