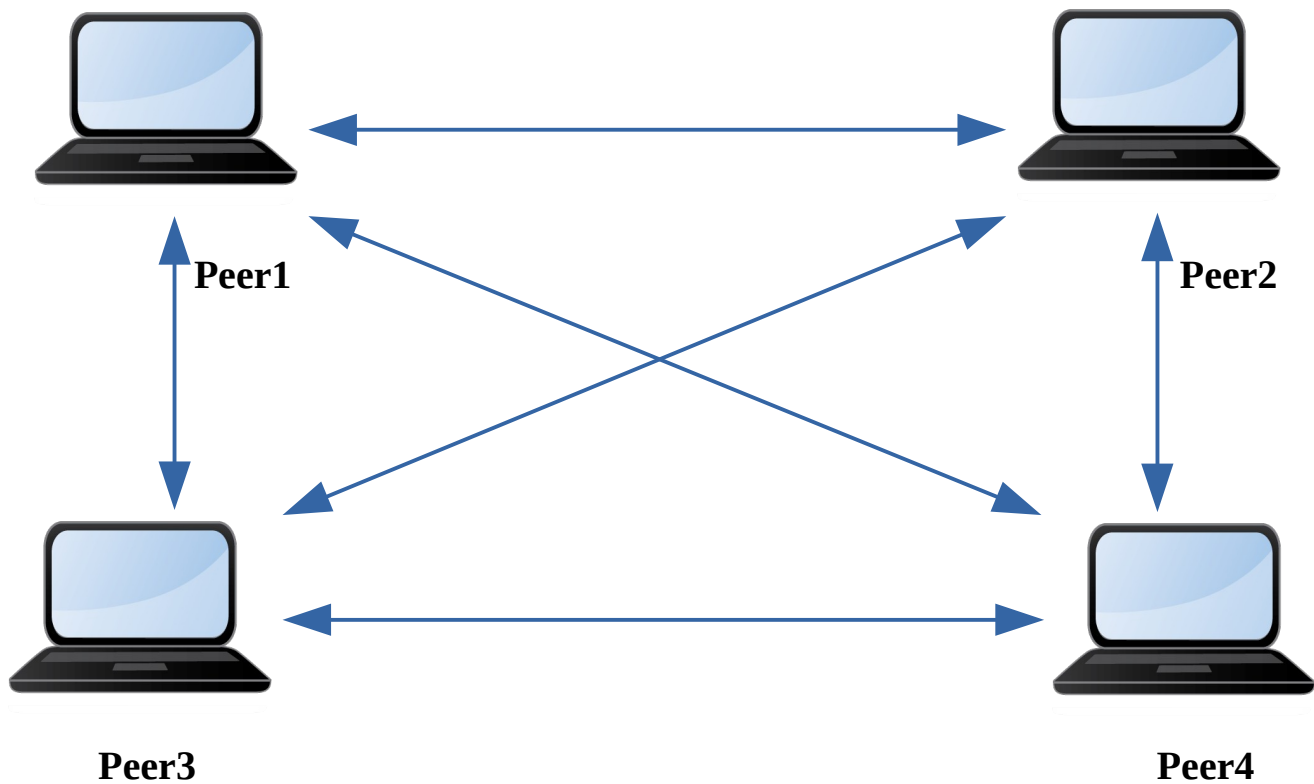


## Design Document :



### **Distributed Hash Table system.**

Its a Distributed Hash table system where all the all the Clients ( Peer ) are connected to all other clients (Peer). Also Each Peer can communicate with another Peer while putting Key-Value pair in hash table Or getting value or deleting value from distributed hash table.

## Classes in Project:



**Client.java**

### 1. Client.java

It's a main class which runs the Client and also start as server to wait for other client requests.

It calls methods of **ClientImpl.java** . Which does three tasks.



**ClientImpl.java**

### 2. ClientImpl.java

It has four methods.

**i. Put :** Accept key value from user, applies hash function on it . Find a server number on which file should be kept and connects to that server to put key value in hash table.

**ii. .Get:** Accept key from client, applies hash function to get server number and retrieves value from that server.

**iii.Delete :** Accept key from client, applies hash function to get server number and delete that value from that server.

**iv.getHashKey :** It Accept key and process it to return an integer which is the server number on which that key-value pair will be stored.

**A List of all sockets and streams are generated and stored in an array. This operation is performed only once.** Then afterwards same streams are used for communication.

### 3. ClientServer.java

Its a simple class which starts client as a server. It starts listening at given port.

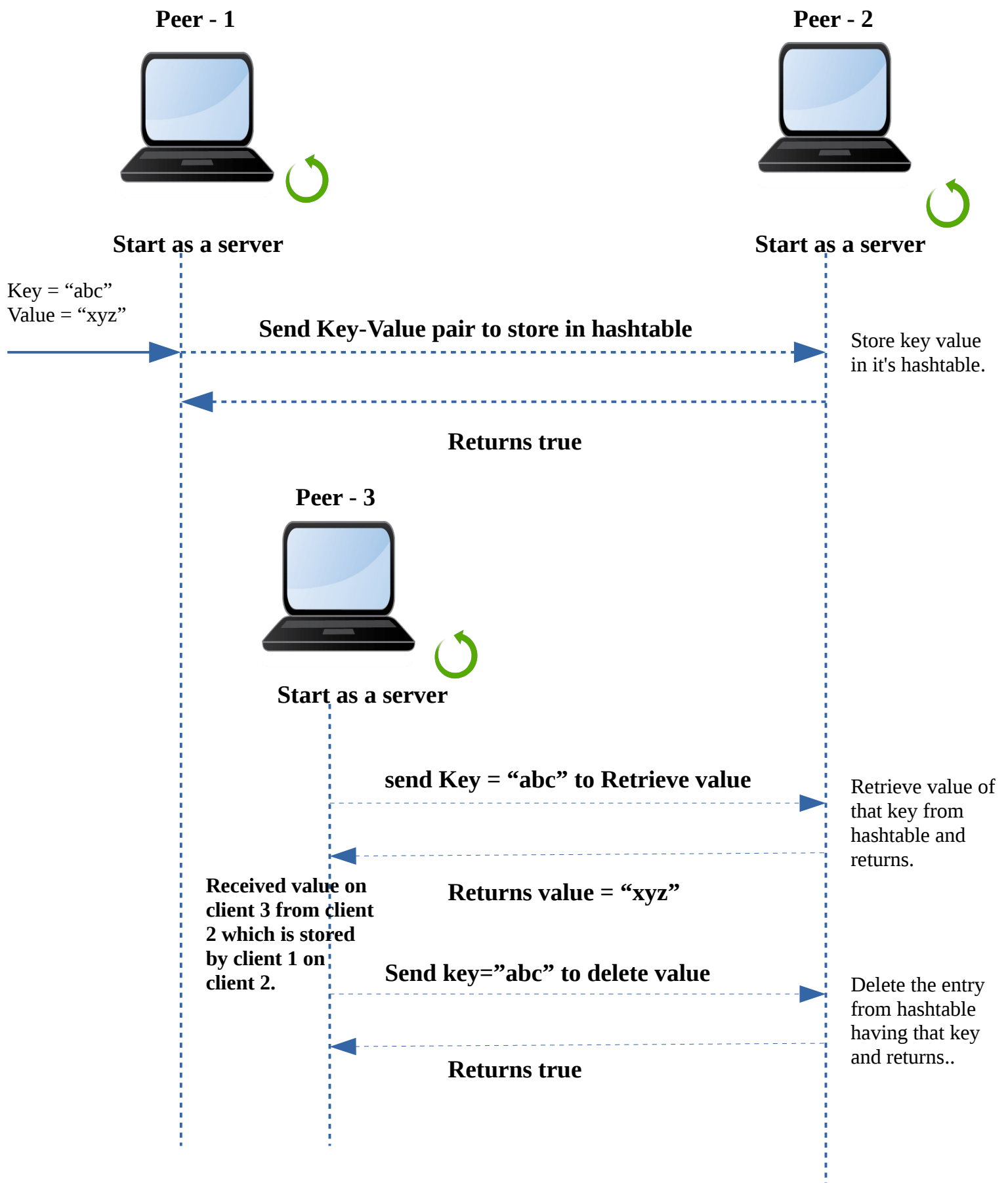
### 4. ClientServerImpl.java

It accepts the key-value and perform Put, Get, Delete operation in its HashTable.

### Overview Of a System:

- When Server gets up, it **creates a ServerSocket connection**. It is **continuously running** and ready to accepts continuous requests from different Client.
- When any Client connects to a server, **first time it accepts the connection and create a new Thread to process a request**. Further all communication is done through same socket on that same thread.
- Here **every client has his own hashtable** where values will be stored. And **all clients has a same HashFunction** which is used to find the server number where that key-value pair will be put. Similarly the same hash function will be used to retrieve / delete records from distributed hashtable.
- **Task** : Its only task is to store key-value pair in system divided hashtable and remove or retrieve entries from those distributed hashtable.

Sequence Diagram :



## **Improvement and Extensions:**

1. **Data Replication/Resilience is not handled.** So the system is not fault tollerant.

**Achieved by :** Replicating the hashtable on some other servers. And if the server is down we should be able to get an Ip address of another Server where hashtable is replicated and should be able to retrieve the record.

2. **If any of the system fails,it leads to fail our distributed hash table system.**