DISTRIBUTED COMMUNICATION SYSTEM

Project Guide
Dr. M Abdul Nizar

Ashitha O M
TVE20MCA-2025

MOTIVATION

- The most popular online chat systems rely on a central server to connect users.
- These services are usually have high up-time, there is no guarantee that service providers will continue to offer their product no matter how Popular
- Online chat service providers can trivially gather personal data such which users chat with each other and at what frequency.

MOTIVATION

- This project provide a completely distributed,P2P chat system based on the Kademlia DHT
- It will remain operational as long as there are people using the system
- No individual person or organization has the ability to take the service offline
- Difficult to track user activity since no single server (or group of servers) is likely to process every request.

DISADVANTAGES OF A CENTRALISED MESSAGING SYSTEM

- Not Free and open-source code
- Use of central server
- Limited bandwidth
- Lesser Security
- Privacy
- Complete failure

LITERATURE REVIEW

CHAT APPLICATION WITH DISTRIBUTED SYSTEM - Shuyang Zhu

> web application for users to instantly communicate with each other.Uses WebSocket for text chat and WebRTC for video and audio and find the result pleasant.

Disadvantage

> MySQL is used to store all the information we want to keep, especially data that does not need to be changed frequently, such as user authentication and authorization information, such as username, password.

LITERATURE REVIEW

TOX CHAT

> Tox began a few years ago, in the wake of Edward Snowden's leaks regarding NSA spying activity. The idea was to create an instant messaging application that ran without requiring the use of central servers.The system would be distributed, peer-to-peer, and end-to-end encrypted

Disadvantage

> Not completely distributed.username things and offline messaging is decentralised.

LITERATURE REVIEW

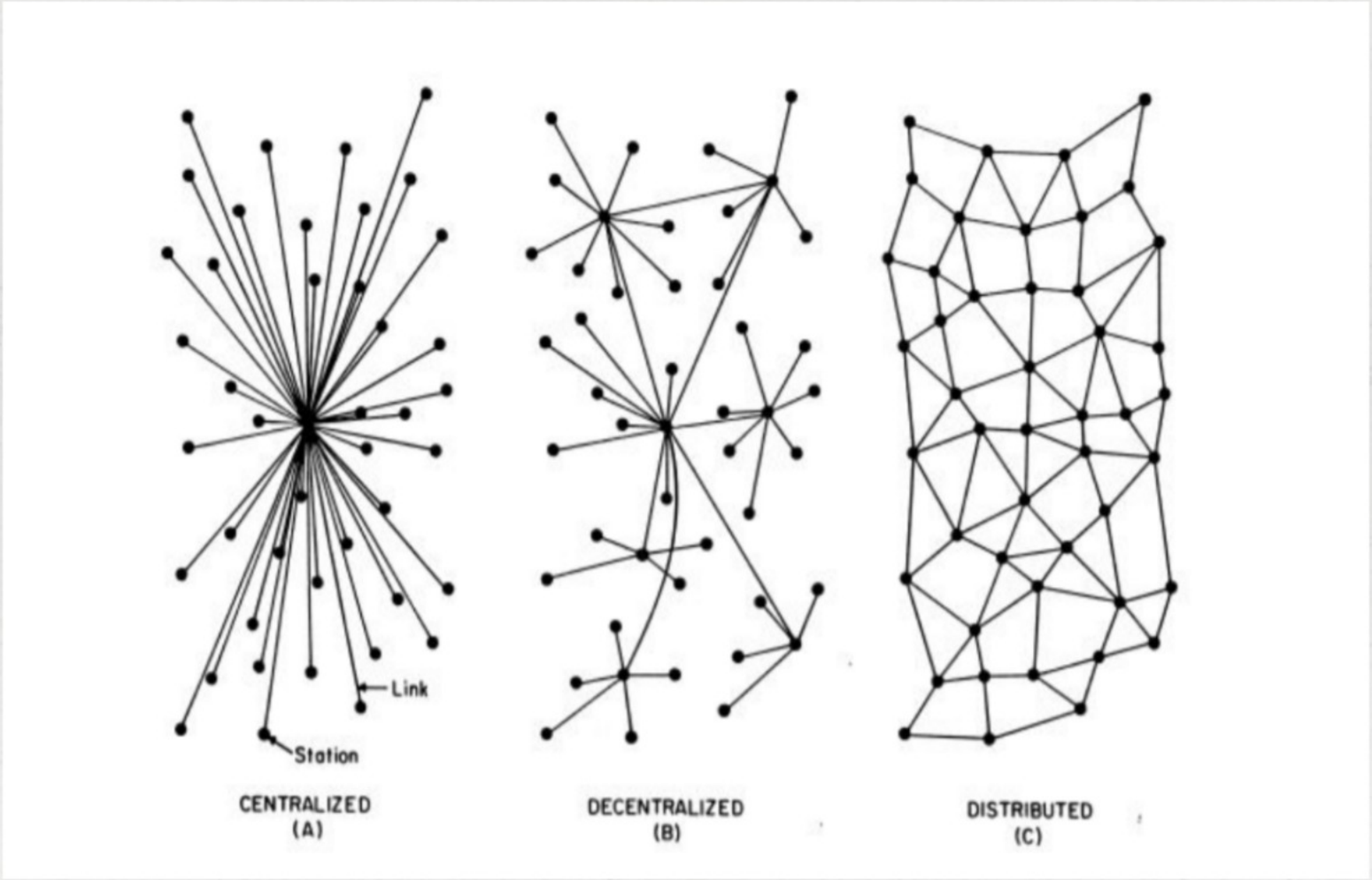
Decentralized Chat Application using Blockchain Technology -Abhishek P. Takale, Chaitanya V. Vaidya, Suresh S. Kolekar

- In this application all the user data is stored on a block which is connected to other blocks forming a chain. As the name suggests, a decentralized application does not have a centralized server. It is basically a peer-to-peer network.

Disadvantage

- Blockchain can slow down when there are too many users on the network.
- Blockchains are harder to scale due to their consensus method.
- High cost

PROPOSED SYSTEM



OBJECTIVE

For the implementation of this project main procedures are the following

- Implement robust index system using a distributed hash table for decentralized chat appications. This indexing system is responsible for storing IP addresses and port of all users once they join the chat
- Implement blockchain name servers for authentication and authorisation with a public-private key pair
- Implement asymmetric cryptography for authenticating users and securing the messages passed between the peers
- Implement a user Interface with GODOT.

OBJECTIVE

Main objectives of this project are

- Privacy:
  - Ensure more privacy by not storing the personal user information.
- Security :
  - Secure the shared data using better cryptography methods.
- Scalability:
  - Lesser complexity for Adding and removing users.

ADVANTAGES OF PROPOSED SYSTEM

- No central server required
- No restriction in bandwidth
- No single point failure
- Higher Security
- Increased privacy
- Works without Internet access

SYSTEM

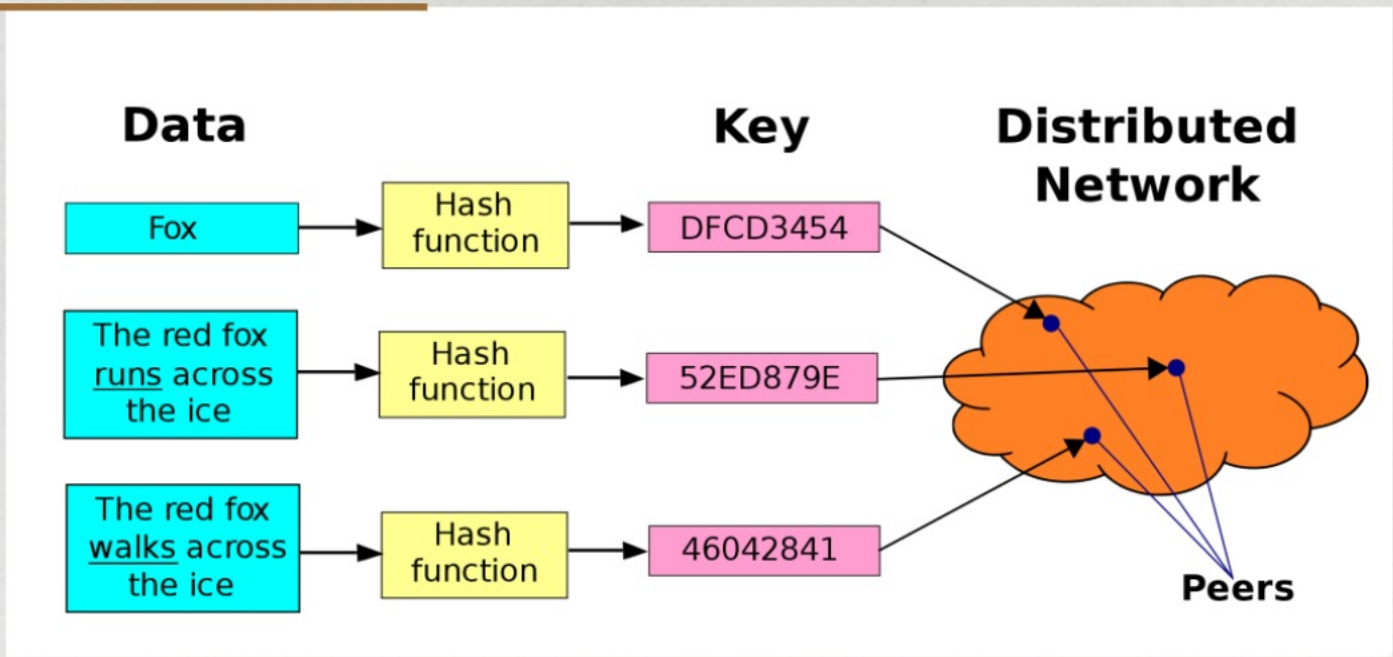
DHT

- A distributed hash table (DHT) is a distributed system that provides a lookup service similar to a hash table
- Nodes can be added/removed with minimum work around re-distributing keys.
- Responsibility for maintaining the mapping from keys to values is distributed among the nodes, in such a way that a change in the set of participants causes a minimal amount of disruption.
- This allows a DHT to scale to extremely large numbers of nodes and to handle continual node arrivals, departures, and failures.

SYSTEM

DHT-

Usage of Distributed HashTable in a network



Main difference between a network such as BlockChain network and DHT lies in the fact that in DHT, not all nodes have complete copy of all data.

SYSTEM

Kademlia algorithm

- One of the famous algorithms to distribute Key-Value pairs is the Kademlia algorithm
- Kademlia is a distributed hash table for decentralized peer-to-peer computer networks designed by Petar Maymounkov and David Mazières in 2002.
- This is very efficient: Kademlia contacts only O(log (n)) nodes during the search out of a total of n nodes in the system.

SYSTEM

Kademlia algorithm

- The Kademlia network is made up of a wide range of nodes. Each node on the network is identified by a unique binary number called node ID. The node ID is used to locate values (block of data) in the Kademlia algorithm. The values are also interlinked within a Kademlia network with a specific value's key, a binary number of fixed length.
- Because of its decentralized structure, Kademlia builds a strong defense against a denial of service attack. Its decentralized structure is equally advantageous when the nodes become flooded

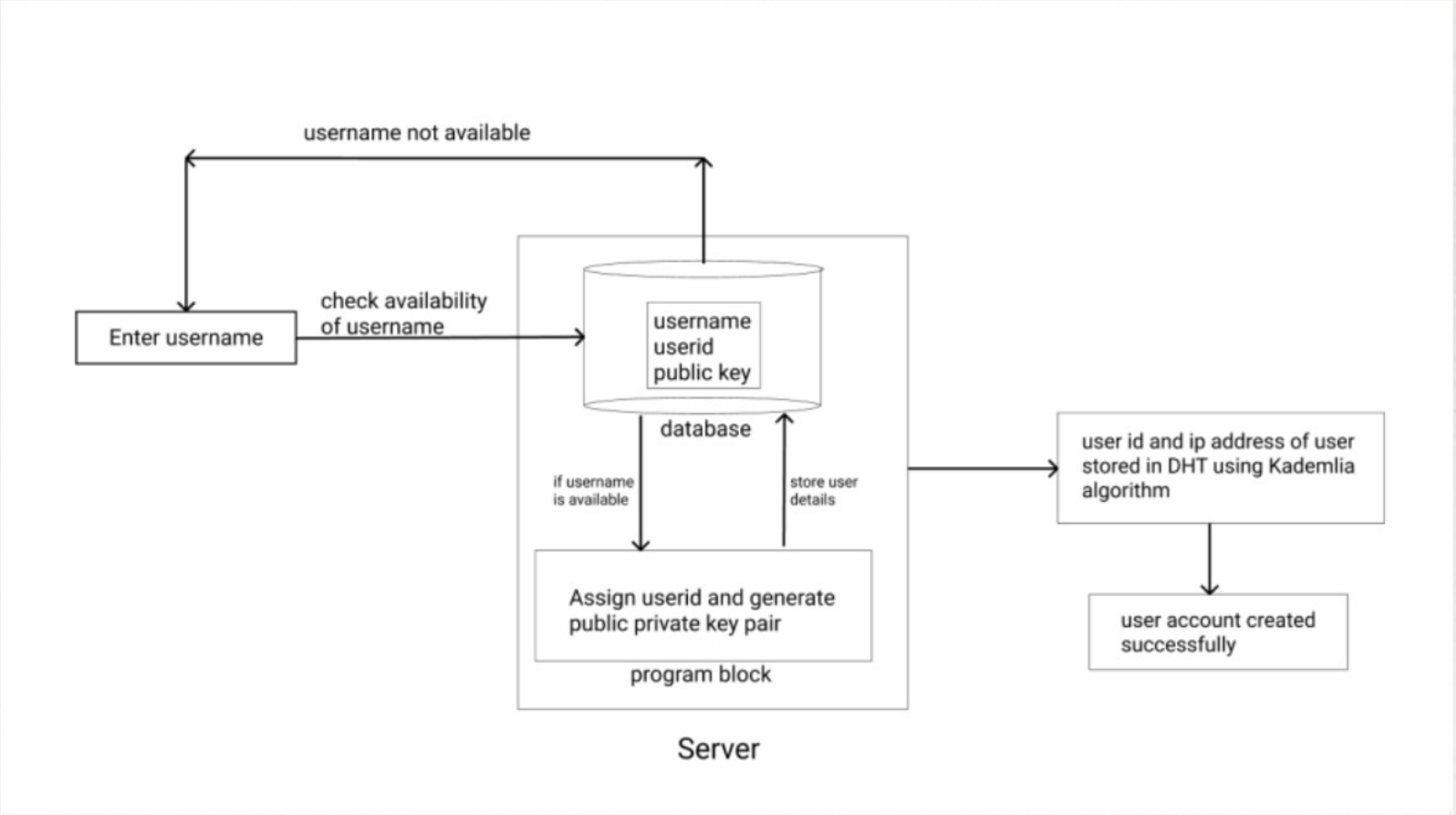
SYSTEM

Golang

- Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.
- 100% utilization of CPU is possible by using Golang. **Golang is a robust system-level language used for programming across large-scale network servers and big distributed systems**. Golang incorporates capabilities like the ease of coding, ease, and efficient code compilation and efficient code execution.

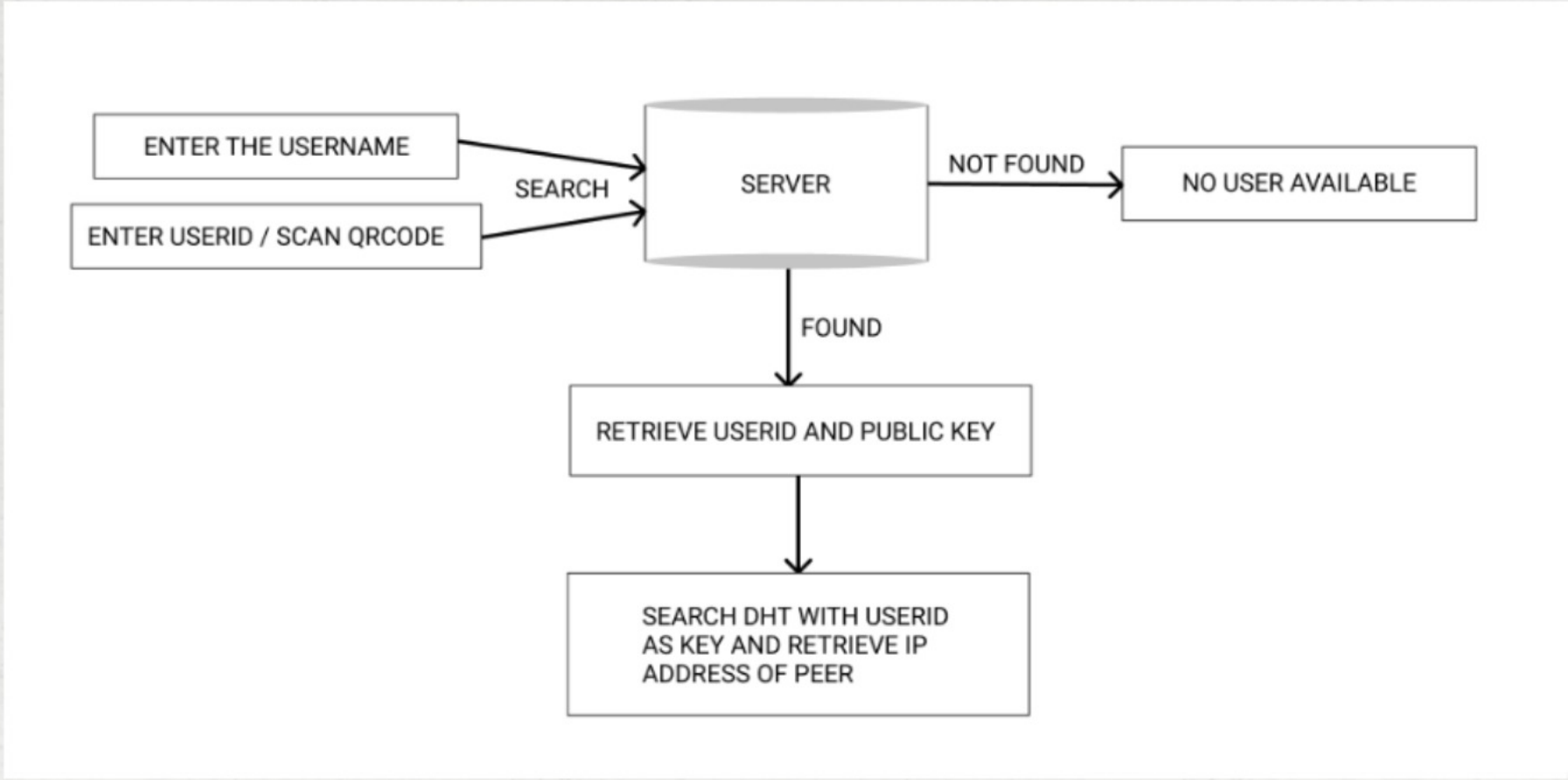
ARCHITECTURAL DESIGN

> USER ACCOUNT CREATION



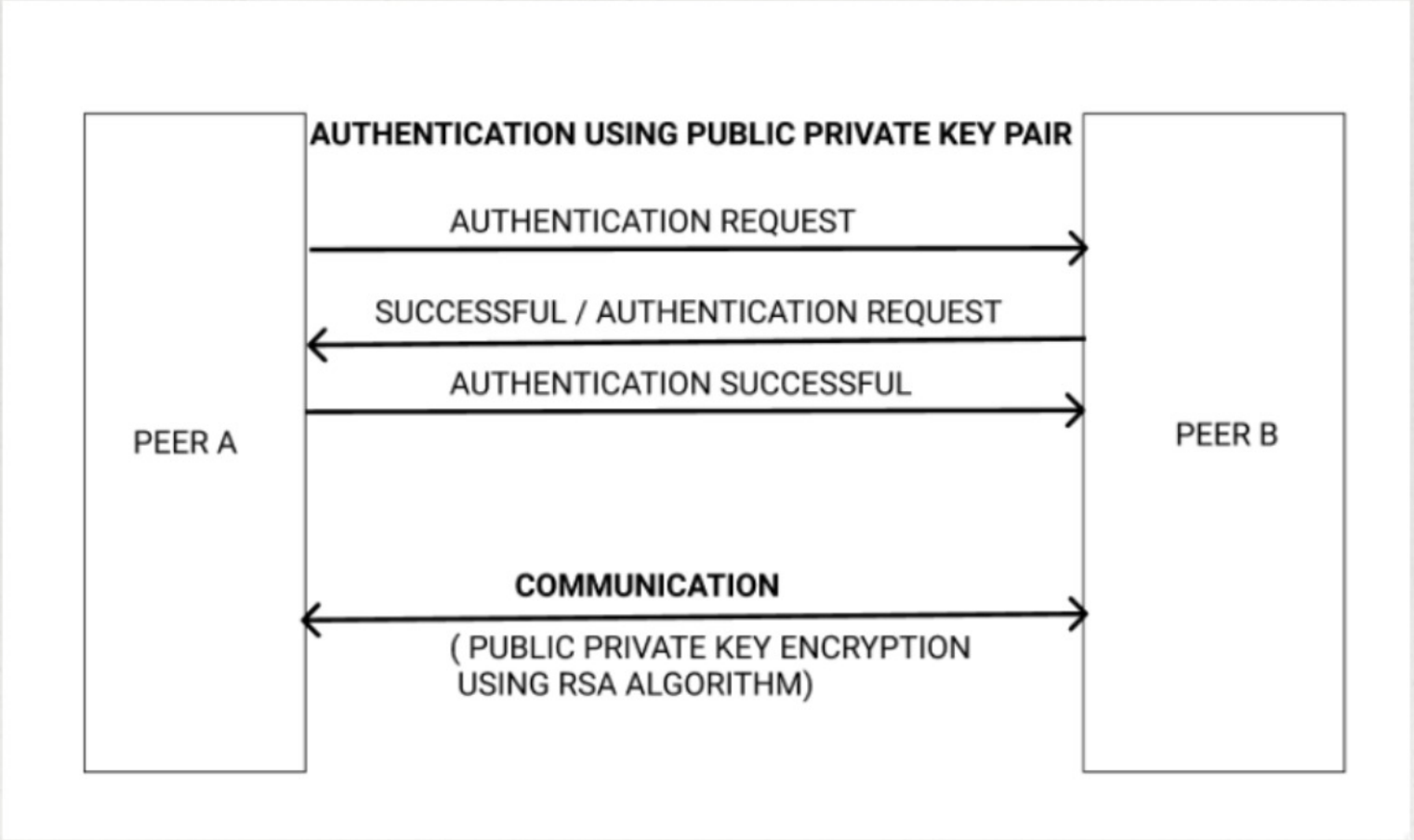
ARCHITECTURAL DESIGN

> FINDING PEER IN THE NETWORK



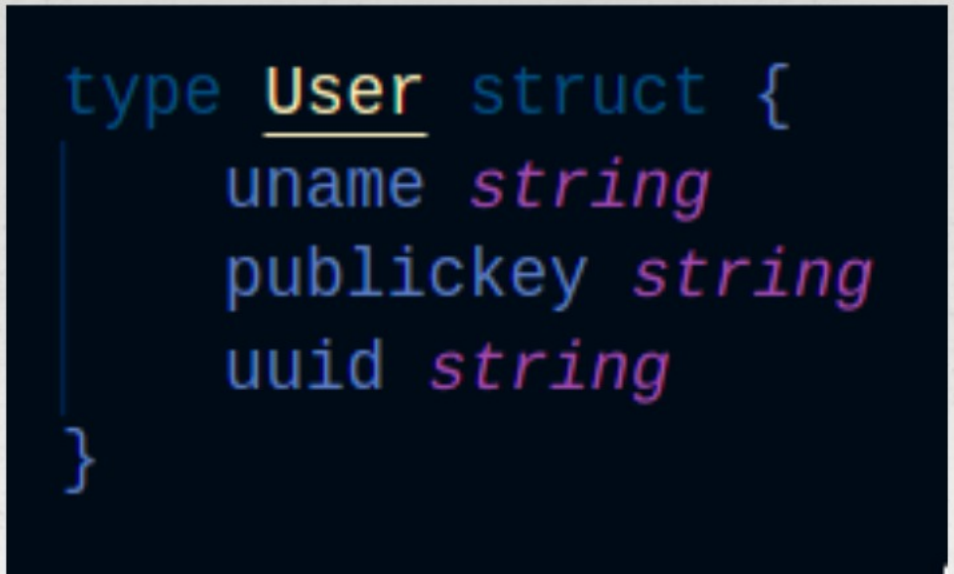
ARCHITECTURAL DESIGN

> AUTHENTICATION AND COMMUNICATION



IMPLEMENTATION

- The user opens the application and creates an account if it doesn't already exist
- The following details are written inside a block
  {
  Userid: a 128 bit long UUID(Universally unique identifier)
  Username: a string of size 30 characters maximum
  Publickey: This is a public key for the user, which is generated on the client side at the time of account creation.
  }



IMPLEMENTATION

- This UUID is created while the account is created on a client device
- First the user makes a request to the dChat name server for the availability of username
- If they are available, the name server requests for other parameters such as Public and userid and writes records corresponding to them in the blockchain.

```
Result {
  '0': 'user_1',
  '1': '7310929070293rs0d09ds8c9sdfg0sdfg09d8fg09s8df0g98sd09fg80s9dfg8',
  '2': 'WWEgdntY8L40B9f1bW0aTrjko6XxJkkpvOvLtAbdFr87YRtoFyOY0F2nSZvpbi3PdZ1AaNfHO6ifQHOFISs0tK
PEtBmzfkMU9lQIISs0tKPEtBmzfkMU9lQIDBfQ7lqIBa23a6WQ0aQPrPABcx3s',
  uname: 'user_1',
  publicKey: '7310929070293rs0d09ds8c9sdfg0sdfg09d8fg09s8df0g98sd09fg80s9dfg8',
  uuid: 'WWEgdntY8L40B9f1bW0aTrjko6XxJkkpvOvLtAbdFr87YRtoFyOY0F2nSZvpbi3PdZ1AaNfHO6ifQHOFISs0t
KPEtBmzfkMU9lQIISs0tKPEtBmzfkMU9lQIDBfQ7lqIBa23a6WQ0aQPrPABcx3s'
}
```

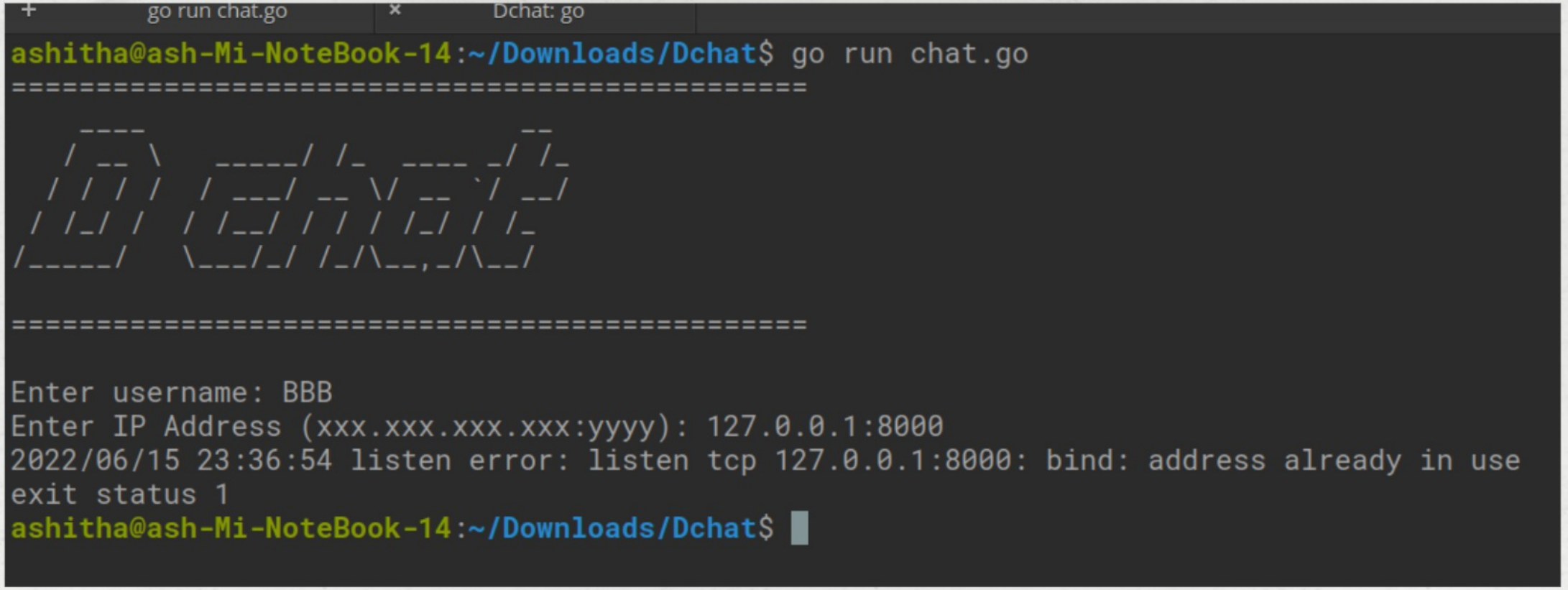
IMPLEMENTATION

```
type DhtNode struct {
    IpAddr string
    NodeId ID // sha1(ip)
    RoutingTable [IDLen][]RoutingEntry // map from NodeId to IP- a IDLen X K matrix
    Kv map[ID]string // map from username to IP
}
```

IMPLEMENTATION

```
ashitha@ash-Mi-NoteBook-14:~/Downloads/Dchat$ go run chat.go
================================================

    ____                        __
   / __ \   _____/ /_  ____ _/ /_
  / / / /  / ___/ __ \/ __ `/ __/
 / /_/ /  / /__/ / / / /_/ / /_
/_____/   \___/_/ /_/\__,_/\__/

================================================

Enter username: AAA
Enter IP Address (xxx.xxx.xxx.xxx:yyyy): 127.0.0.1:8000

[*] Looks like you haven't logged in on this computer before! Would you like to create a new network, or join an existing one?
Join existing? Type (Y/N):N
Connecting to Peerchat...
```

IMPLEMENTATION



IMPLEMENTATION

```
+ go run chat.go × go run chat.go
ashitha@ash-Mi-NoteBook-14:~$ cd Downloads
ashitha@ash-Mi-NoteBook-14:~/Downloads$ cd Dchat/
ashitha@ash-Mi-NoteBook-14:~/Downloads/Dchat$ go run chat.go
================================================
    ____                      __
   / __ \   _____/ /_  ____ _/ /_
  / / / /  / ___/ __ \/ __ `/ __/
 / /_/ /  / /__/ / / / /_/ / /_
/_____/   \___/_/ /_/\__,_/\__/

================================================

Enter username: BBB
Enter IP Address (xxx.xxx.xxx.xxx:yyyy): 127.0.0.1:7000

[*] Looks like you haven't logged in on this computer before! Would you like to create a new network, or join an existing one?
Join existing? Type (Y/N):y

[*] To join an existing network, please enter an IP address/port of a friend (xxx.xxx.xxx.xxx:yyyy): 127.0.0.1:8000
Connecting to Peerchat...
User to chat with:
```

IMPLEMENTATION

```
ashitha@ash-Mi-NoteBook-14:~/Downloads/Dchat$ go run chat.go
================================================

     ____                          __
    / __ \    _____/ /_  ____ _/ /_
   / / / /   / ___/ __ \/ __ `/ __/
  / /_/ /   / /__/ / / / /_/ / /_
 /_____/    \___/_/ /_/\__,_/\__/

================================================

Enter username: AAA
Enter IP Address (xxx.xxx.xxx.xxx:yyyy): 127.0.0.1:8000
Connecting to Peerchat...
```

IMPLEMENTATION PENDING

- After that, our client reaches out to dChat name server to check whether the user account exists or not for the other peer. If it exists, public key and userid is obtained from the NS(Name Server)
- Then the user sends a message to the other node by encrypting the message using the public key of the other node.
- Implement a user Interface with GODOT

REFERENCES

- Spring 4-15-2020
  CHAT APPLICATION WITH DISTRIBUTED SYSTEM
  Shuyang Zhu

- Special Issue - CTRD - 2018
  Decentralized Chat Application using Blockchain
  Technology
  Abhishek P. Takale, 2 Chaitanya V. Vaidya, 3 Suresh S. Kolekar

- Https://tox.chat/

- Kademlia - Kademlia is a distributed hash table for decentralized peer-to-peer computer networks designed by Petar Maymounkov and David Mazières in 2002.

- OpenDHT - An implementation of DHT inspired by Bittorrent mainline DHT