Decentralized Voting System.

PREPARED FOR

Leoforce

PREPARED BY

Hargobind Gupta

hargobindgupta.me

May 22, 2019

1. Project Overview

To design a decentralized application for Indian Voting System.

2. Obstacles

- Votes and rules surrounding this use case can be hacked.
- Byzantine Faults.

3. Why use the blockchain?

Blockchain architecture provides great solutions to many of our key problems:

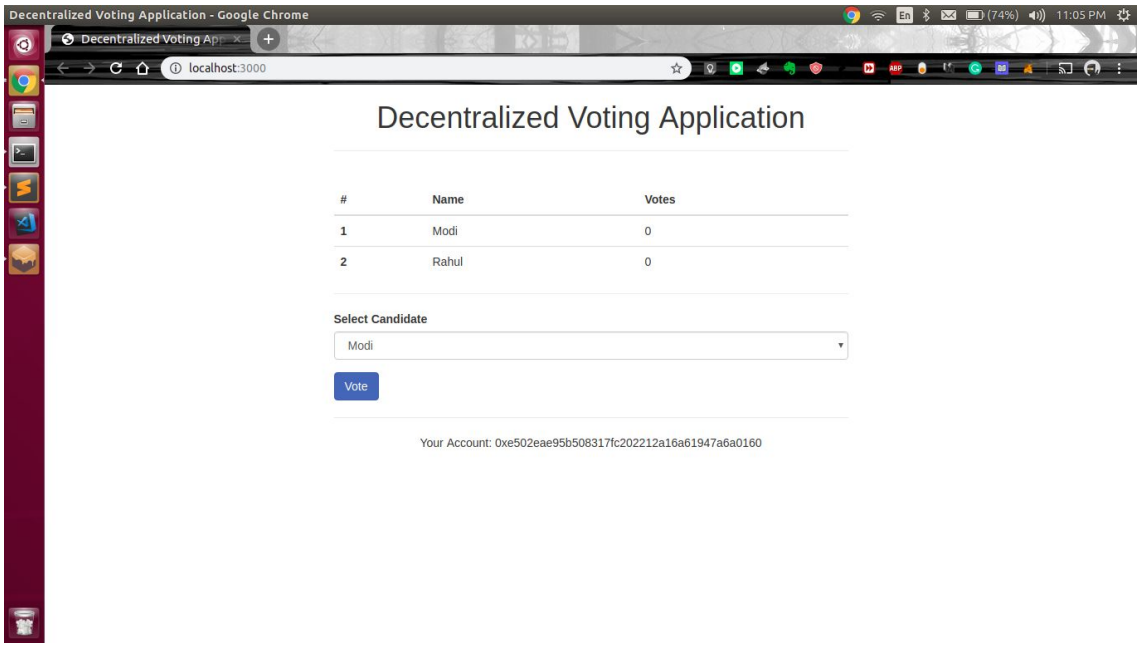
- **High volume**: As voting is decentralized, voting loads are balanced.
- **Data Structure**: Blockchain uses Merkle trees which ensure the robustness of the system, making it difficult to place unauthorized votes or to modify a vote that has already been placed.
- **Decentralized Database**: There are multiple exact replicas of the voting results, created at the time of voting. It would be necessary to fake a vote on all of the redundant servers.
- **Smart Contract:** Rules once coded is decentralized and can not be altered.

4. Dapp

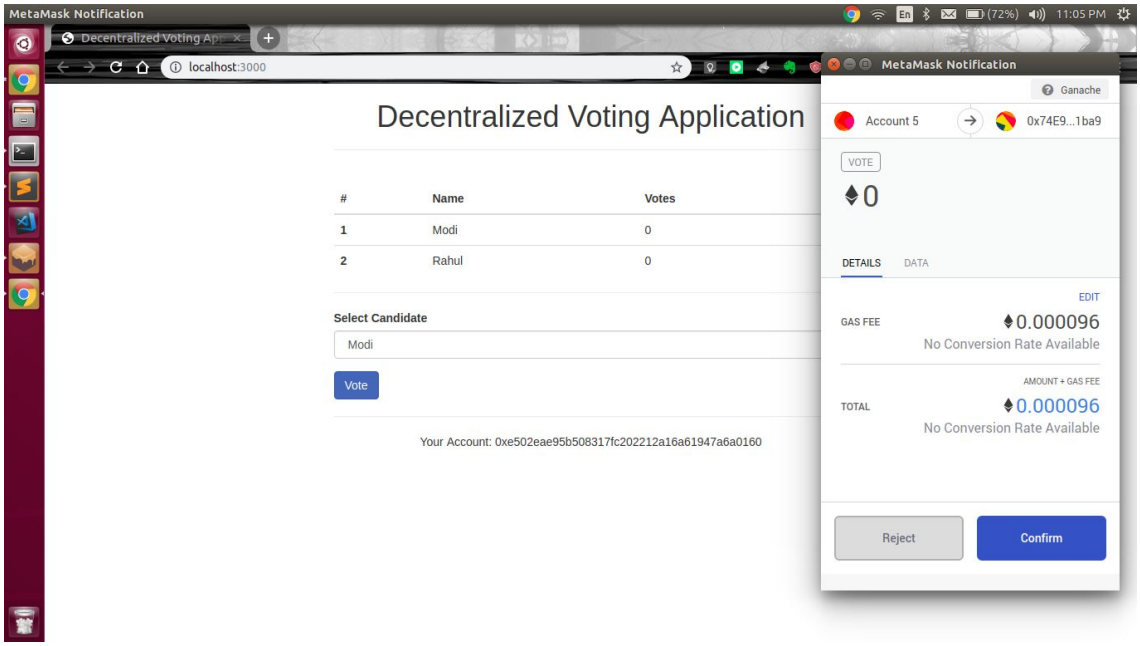
The code is deployed in Github at below link, with a readme file.

URL : https://github.com/Yara1990/voting

Please find the attached Screenshot to get the brief about the Dapp.



2. After selecting a candidate from the dropdown menu, the user clicks on Vote Button, which opens Metamask Plugin to initiate a transaction in order to call the vote function defined in our smart contract.



3.The transaction is confirmed and the functionality to Vode also disappears.This is because a user can only vote once.

Also the count is increased by 1 for Modi.(This info is displayed only for test purpose)