Università della Svizzera italiana

Distributed Systems: Final Project **2022**

Students: Timur Taepov, Filippo Casari, De Grandi Alessandro

Lottery DApp description

1. Introduction

After a long discussion about the possible idea which can be implemented, we decided to develop a decentralized application (DApp) for conducting a lottery. A decentralized lottery allows for a transparent and secure way to participate in a lottery, as it utilizes smart contracts and a decentralized platform such as the Ethereum network. In this project, we implemented the DApp using the Solidity programming language and utilized tools like Brownie and Flask to develop the front-end.

All the instructions about how to run this application on your computer are described at in this github repository.

2. Implementation

2.1. Smart contract implementation

The contract is deployed by the owner of the lottery. The maximum amount of lotteries that can be deployed is bounded by the variable $identificationNumberMax$, for the purpose of tests we set this value to 100. Also, we set a minimum amount of ETH (0.015) that participants can pay to join the lottery. However, players can join multiple times to have more chances of winning. You can see how the contract works on the flow chart [Figure 1].

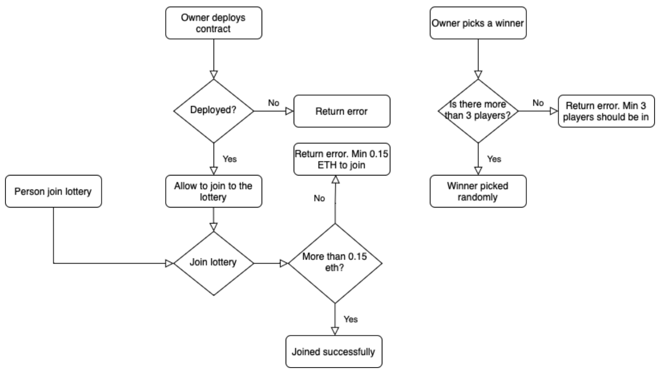


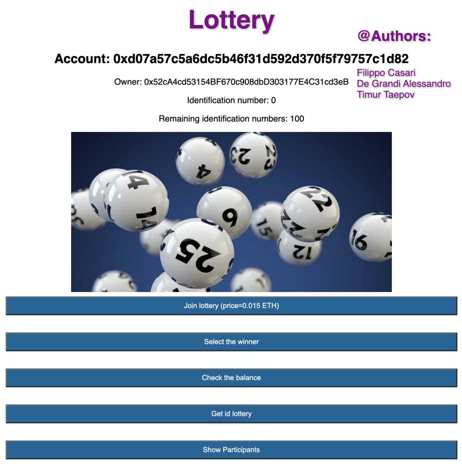
Figure 1: Flow chart

Once the contract is deployed, people can join the lottery. Only the owner of the contract can call the function *selectWinner*() and then the contract randomly selects the winner among the participants using Keccak-256 hash function. The amount of money in the lottery balance is transferred to the winner's ETH wallet.

2.2. Frontend implementation

To implement the front-end we used the following tools:

- Brownie: to compile the solidity contract
- Flask: to implement the web server
- HTML/JS/Css for the GUI



Lottery

@Authors:

Account: 0xd07a57c5a6dc5b46f31d592d370f5f79757c1d82

Filippo Casari
De Grandi Alessandro
Timur Taepov

Owner: 0x52cA4cd53154BF670c908dbD303177E4C31cd3eB

Identification number: 0

Remaining identification numbers: 100

Figure 2: Lottery DAPP

This is how the front end of the application looks like (Figure 2). Here you can see the buttons, whose back end is implemented by contract *Lottery.sol*.

3. Challenges

We faced challenges on how to build a good fronted for our application and how to connect it to the implementation of our contracts. That is why we decided to use Brownie and Flask frameworks to implement this connection in Python.

4. Conclusion

In conclusion, the implementation of a decentralized lottery application has the potential to provide a transparent and secure way for individuals to participate in a lottery. By using smart contracts and a decentralized platform, the process of selecting a winner can be conducted in a fair and unbiased manner. The use of Solidity as the programming language and tools like Brownie and Flask for the front-end development allowed for a functional and user-friendly application.