

# Design Decisions

## Infinity

Infinity is a lottery Dapp using tickets and a fixed jackpot built for the Ethereum blockchain.

Submitted By:

Athul K A  
Arjun P Menon  
Anoop Francis V H

# 1 Introduction

## **1.1 Document Scope**

This document shares the information about how we came to this idea of developing software & what is the study we made behind before we started making the application. All the technologies we chose and the process flow of the application. How the application makes use of the Blockchain technology efficiently so that a full-fledged application can be used for Lottery and fair prize distribution.

## **1.2 Project Overview**

We are building a Blockchain based Dapp on top of Ethereum Blockchain which depicts a Lottery and decentralized prize distribution. That means the people who are interested to play a little luck can buy tickets and win rewards . The application using most of the features of Blockchain which make it Transparent to all audience, Distributed and Tamper-proof. We have developed the application with basic HTML for front-end and Blockchain back-end. We used Solidity programming language to do all the Blockchain based operations.

The application got 4 main tabs, Ticket Buying, Metamask registration, Selection of Winner and finally price distribution. These help the users to play Lottery smoother and easy. The UI is really simple so that anyone even if he is not into computers can operate with ease. We hope this project will change the present Lottery systems-clients to an all new level.

Also, we are sharing the GitLab repository for this application here.

Link: <https://gitlab.com/shscs911/infinity>

## 2 High-Level Design

### 2.1 Goals and Guidelines

The Goals & Guidelines of our project is described below: The present system of Lottery we have are making us pay for the ticket and no further details on that is provided, we don't even know where is the money going and how it is utilized for price distribution. It's completely a centralized system & the details are not transparent.

- Since everything is blockchain based, we can ensure that this is secure in nature and transparent since all the people using the application can see the total tickets distributed to each individual which makes it worth to give a try and play it in a wise manner.
- There is no central system, Money from the user is directly send to the contract. No middle man, User is happy in a such a Lottery which gives out the price in a proper way.

### 2.2 Architectural Strategies

- We decided to pursue the lottery Dapp taking into account Blockchain properties like distributed ledger technology and immutability of the blockchain.
- A set of fixed number of tickets is made available(here 5) and players buy the tickets for a fixed amount and the draw takes place when only the whole tickets are brought this way there will be a sure winner.

Now there were two options to do

- ✓ By adding address to an array and then selecting the random number and placing it as the index location.
- ✓ By making a mapping ticketToOwner where the idea was to assign a number each time a ticket is bought (paying ether to selection function). and the random number is mapped back to the address.

We went with the idea of array setting the number of tickets to be fixed and making a static array of address where the player address is stored and from were the winner address is selected and for making a future proposal of giving second and third price, since array is iterable.

The single winner mapping was a suitable option:

- ◆ We went with the specific solidity compiler version 0.4.25 and truffle 4.1.15 for making the transfer function work because the same was causing error of `infinite gas` on execution in the latest versions.

#### The Contract works in a simple way:

- Player buys ticket using his metamask account.
- The already bought ticket can't be purchased again since it checks a condition of whether there is an address in the index or not.
- The draw happens when all the tickets are bought. (from different accounts is favoured)
- A random number is generated based on blockinfo
- The winner is selected by taking the random generated number into the index of participant array.
- Prize is distributed where the winning address gets the total amount and a reset function is called to reset the array.

## **2.3      *High-Level Deployment***

Below describes how to deploy the application in an environment.

- Installing Infinity to use on your machine is simple. First clone the repository by

```
git clone
```

- Next, navigate to the cloned directory and install the necessary Node.js modules with

```
npm install
```

- Create and place your 12-word mnemonic private key at the location

```
keystore/eth_keystore.txt
```

- Initialize a local blockchain with

```
ganache-cli
```

- Then deploy the smart contract with

```
truffle migrate
```

- Finally, to open the web app,

```
npm run dev
```

### 3 Glossary

**Solidity:** Solidity is a contact-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

**Web3.js:** web3.js is a collection of libraries which allow you to interact with a local or remote ethereum node, using an HTTP or IPC connection.

**Truffle:** A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier.

**Ganache:** A personal blockchain for Ethereum development you can use to deploy contracts, develop your applications, and run tests. It is available as both a desktop application as well as a command-line tool (formerly known as the TestRPC)

**Node.js:** Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.