ZIGURAT

INNOVATION & TECHNOLOGY BUSINESS SCHOOL

2021/2022

FINAL MASTER THESIS

Technical Path Option 2

Coding Smart Contracts on a Blockchain

SIEGFRIED BOZZA

Executive Summary 4
Introduction 5
I. Coding a Decentralized Application...6
I.1. Code 6
I.1.1. Smart Contracts 6
1.1.1 Lottery Smart Contract 6
1.1.2 Lottery Token Smart Contract – Lottery ERC20 Token...7
1.1.3 ChainLink VRF Oracle – A Verifiable Source of Randomness 7
1.1.4 Compound V3 – Lending USDC to Generate Interests 8
1.1.5 USDC – ERC20 Token 8
I.1.2. Front end & DApp 8
I.1.3. Notes on Hardhat and Ethers.js...9
I.2. List of features 10
1.2.1. Overview 10
1.2.2. Entering the Lottery..10
1.2.3. Chainlink Keepers – Automation Overview 11
1.2.4. Picking the Winner..13
1.2.4.1. Checking the end of play-time & Requesting a random number..13
1.2.4.2. Using the random number to select the Winner 14
1.2.5. Calculating the Winner Gains 15
1.2.6. Withdrawing from the Lottery...17
1.2.7. Ending a Lottery round – Lending...18
1.2.8. Admin functions..19
II. Business Case 20
II.1. Problem...20
II.1.1. Central authority means single point of failure...20
II.1.2. Traditional lotteries, a one winner and all other losers game model 20
II.1.3. Traditional lotteries and the transient lottery ticket value 20
II.1.4. Traditional lotteries, centralized governance model and passive players 20
II.2. DApp/Solution 21
II.2.1 Decentralization for a provably truly randomness and incorruptible lottery 21
II.2.2. Decentralized no-loss lotteries, integrating DeFi products, a game changer in the gaming world 21

II.2.3. Lottery, perpetual chance to win, and ticket value beyond the chance to win........................21

II.2.4. Decentralized lotteries and the power of DAOs..22

II.3. Unique Value Proposition...22

II.4. Customer Segments..22

II.5. Unfair advantage..22

II.6. Channels...23

II.7. Key Metrics...23

II.8. Cost Structure...23

II.9. Revenue Streams..23

Conclusion...24

Declaration..26

References...27

Executive Summary

This report draws the opportunity for using decentralized technologies to revolutionize the traditional lotteries.

Indeed, traditional lotteries suffer from several drawbacks for the players.

1. By essence, and despite the laws and regulations aiming at protecting the players, these old lotteries schemes, ruled by a central authority (the lottery organization), relies upon players' trust for the lottery to behave honestly, which constitutes a single point of failure;
2. Secondly, in these old lottery schemes, the only true winner is always the lottery itself: at every lottery round, only one winner (or a few, depending on the lottery set up) is picked up to win a prize. All other players lose the money they invested in buying a lottery ticket;
3. Thirdly, for getting a chance to win, a participant needs to buy a lottery ticket for every lottery round, as a ticket is valid only for a round;
4. Thus, these lottery tickets hold only a transient value, the transient chance to win a prize;
5. Finally, these old lotteries systems are solely governed by their board of directors, and the players are not invited to participate in their governance.

Aiming at solving these issues, the current report describes a series of Smart Contracts and decentralized systems, along with their interactions, allowing to build a decentralized lottery system, benefiting from:

1. A trustless environment where all rules are transparent to all participants, and enforced programmatically, by deploying the Lottery Smart Contracts on the public *Ethereum* blockchain, and by using a decentralized proven source of randomness;
2. A fair prize distribution, where there is no loser: the winner wins the gains, and all other players do not lose their investment, using DeFi protocols to generate the gains;
3. A Lottery ticket whose life spans across multiple lottery rounds, allowing loyal players to reuse their ticket(s), as long as they keep their funds in the Lottery;
4. The possibility for the Lottery tokens, as being issued as ERC20 tokens, to see their value appreciate over time, potentially generating a second source of value for the players, besides the lottery gains;
5. The possibility to build a community of players, sharing the common goal of further improving the Lottery and building upon it, by using the Lottery Tokens as governance tokens, allowing players to participate actively in the Lottery governance decision-making process.

The *no loss* Lottery described in this report allows participants to contribute with cryptocurrencies to a shared Lottery pool, that reinvests funds in a yield-generating DeFi product, and then pays out the interests as rewards to the lucky 'ticket' holder, at every Lottery round. Unlike a traditional lottery, our set of Smart Contracts allow players to recuperate their investments if they lose the draw, or to keep their funds in the Lottery to reinvest in the next Lottery rounds. Thus, a decentralized lottery built upon the rules described above, has the power to enlarge the scope of traditional lotteries, by inciting not only usual chance-based game players, but also crypto investors, to play our game. Moreover, as opposed to traditional lotteries, besides their pure financial advantages, decentralized lotteries that make use of DAO features also offers an environment where participants can collaborate, to build and improve an ecosystem, empowered by their shared values.

Introduction

This report describes the development and implementation of our decentralized no loss Lottery.

As mentioned in the *Executive Summary*, our Lottery offers several advantages compared to the traditional lottery schemes, by offering:

1. A trustless environment, in which immutable rules (written in the Smart Contracts code) are deployed on the *Ethereum*[i] public blockchain, allowing an enforced execution of the code, accordingly to the rules;
2. A reliably proven decentralized source of randomness to generate true random numbers, and thus fairly pick up the winners, using the *Chainlink VRF* oracle services;
3. A fully decentralized automation of the Lottery system, using the *Chainlink Keepers – Automation* oracle services;
4. A *no loss* set of mechanisms, using the *Compound* DeFi protocol, allowing to lend the players' deposits to generate interests, used to reward the Lottery winners;

The following sections describe how we can build a Lottery set of Smart Contracts that is secure, perpetual, provably random, generating interests, and offering *no loss* and only gains.

I. Coding a Decentralized Application

In the *I.1. Code* section below, we describe the Smart Contracts built and used in our Lottery project, along with the decentralized mechanisms used to generate a random number to pick up a winner (*Chainlink VRF*).

In the *I.2. List of features* section, we will then describe in more details the features of the Lottery Smart Contract, how it interacts with the other contracts, and the decentralized mechanisms used to automatically switch the Lottery states (*Chainlink Keepers – Automation*).

I.1. Code

The Smart Contracts used for our Lottery platform are written using the *Solidity*[ii] programming language, and are deployed on the *Ethereum Goerli*[iii] Test network.

The *Goerli* Testnet is the first proof-of-authority cross-client testnet, synching Parity Ethereum, Geth, Nethermind, Hyperledger Besu (formerly Pantheon), and EthereumJS. Born in September 2018, this testnet is a community-based project, completely open-source.

In a future implementation of the Lottery, it is anticipated that our smart contracts will be deployed on the *Polygon*[iv] blockchain, for allowing players to experience faster and cheaper transactions, while still benefiting from the *Ethereum* base-layer network security.

I.1.1. Smart Contracts

We will review below the Smart Contracts used to implement our Lottery platform, with links to their repositories and the associated documentation.

1.1.1 Lottery Smart Contract

The Lottery Smart Contract has been designed as the core of the on-chain business logic of the Lottery project.

The code of this contract can be found on this Github repository

A deployed (Goerli) and verified Lottery Contract instance can be found on Etherscan

As on overview, this contract is responsible for:

- Deployment of the Lottery Token (*LTK*) Smart Contract;
- Interaction with the Players, and with the *LTK*, *USDC*, *Chainlink VRF*, *Chainlink Keepers*, *Compound* set of Smart Contracts;
- Management of Players' *LTK* and *USDC* balances;
- Management of the time given for each Lottery round to be entered;
- Management of the time given after each Lottery round for the players to eventually withdraw their funds; as described in the next sections, this "*time management*" is performed through interaction between the Lottery Contract and the *Chainlink Keepers* oracle services;
- Calculation of the winner's address and gains;
- Management of the *USDC* lending and withdrawal on the *Compound V3* protocol.

The repository mentioned above contains detailed notes on the main variables and functions used.

1.1.2 Lottery Token Smart Contract – Lottery ERC20 Token

The LTK Smart Contract is used to mint the LTK tokens associated with the Lottery. It manages the distribution and ownership of the LTK tokens among the players.

A deployed (Goerli) LTK Contract instance can be found on Etherscan

This Contract is inheriting from the ERC20 token standard, using the [v]￼ library. *OpenZeppelin* offers tested libraries of Smart Contracts for *Ethereum* and other blockchains, and includes the most used implementations of ERC standards, like the ERC20.

During the deployment of the Lottery Contract, the LTK Contract is deployed, *x* number (passed to the constructor function) of LTK are minted, and their ownership is set to be the Lottery Contract itself. Once deployed, the Lottery Contract Owner has the possibility to mint fresh LTK at will.

1.1.3 ChainLink VRF Oracle – A Verifiable Source of Randomness

For a Lottery game to be a true game of chance, the mechanism used to pick up the winner must be totally unpredictable, and reliable over time.

Different solutions[vi] have been attempted on-chain, for instance using the block hash of the current block, as a seed to generate a random number. But because a Blockchain is a deterministic system, we cannot create truly random numbers directly from a smart contract on-chain, but only *pseudo-random numbers*, repeating themselves after a particular sequence.

Another solution might be considered, off-chain, using an API to get random numbers from a centralized oracle, but with the risk of bringing a single point of failure in our application.

Thus, as a secure and fair source of randomness to pick up the Lottery's winner, our Lottery game is making use of the decentralized *ChainLink Verifiable Random Function*[vii] (*VRF*) oracle services, renowned as the industry standard *random number generator* (RNG) solution for smart contracts and off-chain systems that require a cryptographically secure, transparent, and provably fair source of randomness. It provides verifiable randomness to smart contracts across multiple blockchain networks, including Ethereum, Polygon, and Binance Smart Chain.

Using the *Chainlink VRF* oracle, a request for a random number is sent to multiple Chainlink nodes. Each node generates a random number in a verifiably random fashion, using its public and private keys to cryptographically prove that the number was random. This random number generation, done across multiple nodes, guarantees that there is no single source of failure, and then all answers are combined (using the bitwise *XOR* operator) to generate the final random number.

Each random result is thus verified on-chain with cryptographic proofs, so that malicious users, node operators, and even the Lottery smart contract admins cannot tamper with the resulting randomness.

Our Lottery contract, by importing the *VRFConsumerBaseV2* and *VRFCoordinatorV2Interface* contracts interfaces, is able to interact with the *Chainlink VRF* system.

This system allows our Lottery contract to programmatically send a request for a provably-fair and verifiable random number, at the end of each Lottery *play-time* round, providing our players with a trustworthy experience.

1.1.4 Compound V3 – Lending USDC to Generate Interests

Part of the *DeFi* ecosystem, *Compound* is a decentralized, blockchain-based protocol that allows users to lend and borrow crypto assets, and have a voice in its governance, using its native *COMP* token.

More specifically, the *Compound V3* [viii] protocol, used in our Lottery, is an EVM compatible protocol that enables crypto assets supplying as collateral, in order to borrow the base asset. Accounts can also earn interest by supplying the base asset to the protocol. The initial deployment of *Compound V3* is on *Ethereum* and the base asset is USDC. As the loans taken by borrowers are over-collaterized, the lenders are guaranteed to generate interests on supplying the base asset.

The *Compound V3* set of smart contracts are deployed on the *Ethereum Main net*, and on the *Goerli* and *Avalanche Fuji test nets.*

Our Lottery contract, by importing the *IComet* contract interface, is able to interact with the *Compound V3* system, making our Lottery able to lend the players' USDC deposits, in order to generate interests and hence the gains for the winners.

As a starting implementation, our Lottery platform is using the *Compound V3* protocol to lend the base asset USDC. It is anticipated that further developments of our Lottery will make use of the borrowing functions of the *Compound V3* protocol, to potentially increase the gains for the Lottery winners.

1.1.5 USDC – ERC20 Token

Originally launched on a limited basis in 2018 by two founding members, the peer-to-peer payment services company *Circle* and the *Coinbase* cryptocurrency exchange, the USD Coin (ticker USDC[ix]) is a stablecoin, pegged to the U.S. dollar on a 1:1 basis. Every unit of this cryptocurrency in circulation is backed up by $1 that is held in reserve, in a mix of cash and short-term U.S. Treasury bonds.

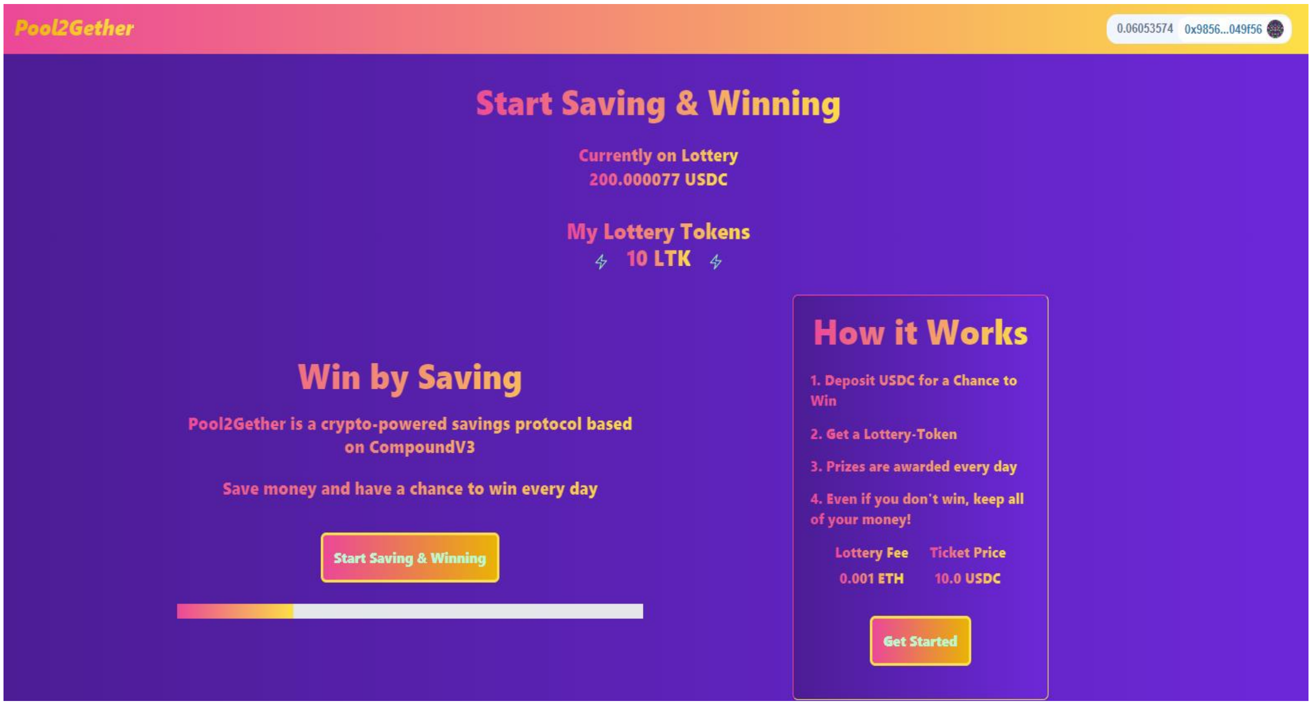
The USDC Smart Contract imported in our Lottery is the one documented by the Compound V3 protocol. This USDC contract is based on the ERC20 token standard, and can be found on Etherscan

I.1.2. Front end & DApp

A functional implementation of the Lottery smart contract has been developed, and the live DApp (on the Goerli test network) can be found on Pool2Gether

The Lottery front-end was built around a Next.JS App. Containing detailed notes about the functions used, the code can be found on this Github repository

Interestingly, the Lottery Contract emitted *Solidity* events are used to trigger the front-end update.



Start Saving & Winning

Currently on Lottery
200.000077 USDC

My Lottery Tokens
10 LTK

Win by Saving

Pool2Gether is a crypto-powered savings protocol based on CompoundV3

Save money and have a chance to win every day

How it Works

1. Deposit USDC for a Chance to Win

2. Get a Lottery-Token

3. Prizes are awarded every day

4. Even if you don't win, keep all of your money!

Lottery Fee	Ticket Price
0.001 ETH	10.0 USDC

Figure 1: Lottery DApp screenshot

I.1.3. Notes on Hardhat and Ethers.js

HardHat[x] , a development environment for Ethereum software, was used to create, test, and deploy our Smart Contracts. As the project, at the time of writing, is still in development, the tests were written on different Lottery implementations and are not yet fully integrated.

Ethers.js[xi] is used in our front-end code, as a library for interacting with the Ethereum Blockchain and hence, with the Smart Contracts.

I.2. List of features

1.2.1. Overview

Each Lottery round is divided into 4 time intervals, the Lottery being subsequently in an *Open to Play*, *Calculating the Winner*, *Calculating the Winner Gains*, and *Open to Withdraw* state. These states are monitored and managed by the *Chainlink Keepers* oracle service, as described below.

Detailed notes are available in the code on this [Github repository]

Our Smarts Contracts implement the following features and user experiences.

1.2.2. Entering the Lottery

If the Lottery is *Open to Play*, a user can enter the Lottery by fulfilling 3 steps, each transaction being initiated from the front-end:

- Sending the correct amount of USDC to buy one Lottery Ticket;
- Giving allowance to the Lottery Contract to manage the player's LTK tokens ownerships. This allowance is necessary in further stages, when a player interacts with the Lottery Contract to withdraw its funds, in which case the Lottery Contract not only transfers the USDC due amount to the player, but also transfers the ownership of the player's LTK tokens to itself;
- Sending the correct amount of ETH to access the Lottery.

For each Lottery ticket bought, the user receives one LTK token.

Immediately upon entering the Lottery, the USDC amount transferred by the user to the Lottery, is transferred to the *Compound V3* protocol for lending, starting to generate interests.

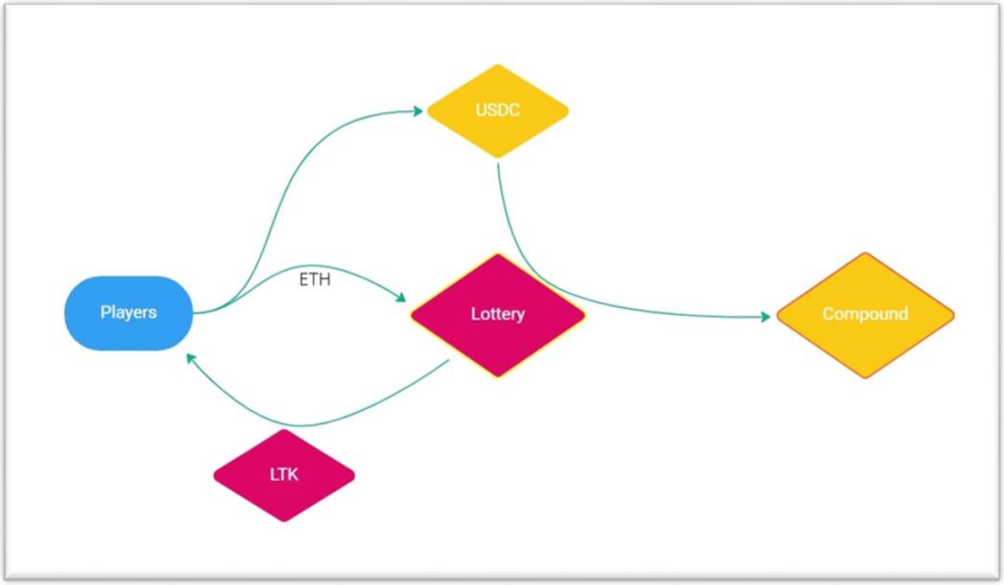


Figure 2: Entering the Lottery

By entering the Lottery game, the first player of a lottery round sets the date for the end of this round. The *Chainlink Keepers* nodes are responsible for automatically triggering a cascade of events, as described below.

1.2.3. Chainlink Keepers – Automation Overview

The *Chainlink Keepers-Automation*[xii] enables conditional execution of Smart Contracts functions. The triggers can be either programmatic (*Custom logic triggers*) or *time-based*.

Custom logic triggers allow Developers to provide custom *Solidity* logic that *Automation Nodes* evaluate (off-chain) to determine when to execute the functions of a Smart Contract, on-chain.

The architecture of the *Chainlink Automation Network* contains three main actors:

- *Upkeeps*: the tasks that Developers want to execute on-chain, if a specific set of conditions are met;
- *Automation Nodes*: Nodes in the *Chainlink Automation Network* that service registered;
- *Automation registry*: governs the actors on the network and compensates *Automation Nodes* for performing successful *Upkeeps*. Developers can register their *Upkeeps*, and *Node Operators* can register as *Automation Nodes*.

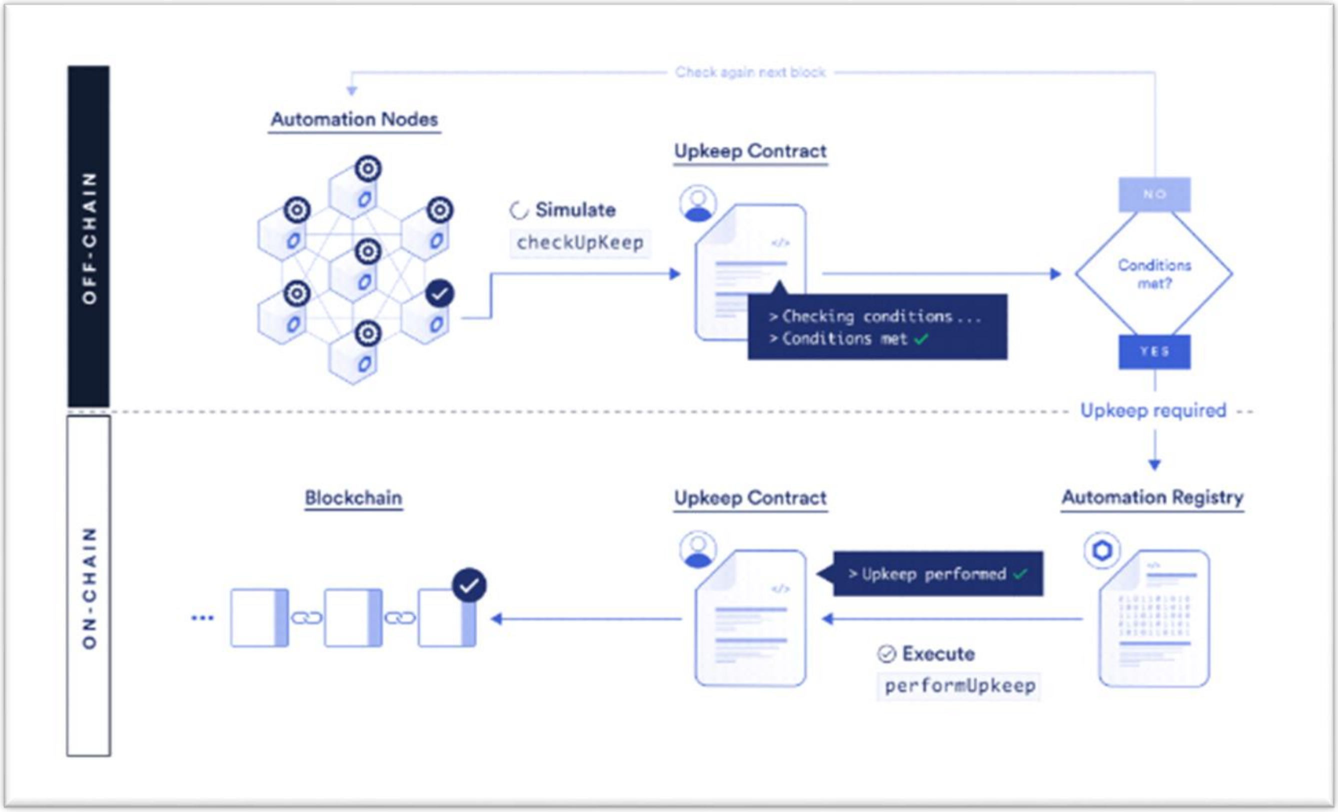


Figure 3: Chainlink Keepers - Automation

During every block, the *Automation Nodes* review all of the *upkeeps* to determine which ones are eligible. This check is done off-chain using a *geth* simulation. The *Automation Node* checks both the *checkUpkeep* and *performUpkeep* conditions independently using simulation. If both are true (eligible), and the *upkeep* is funded, the *Automation Node* proceeds to execute the transaction on-chain.

To benefit from *Chainlink Keepers* services, the Lottery Contract imports the *KeeperCompatibleInterface* contract interface and must contain two functions:

- *checkUpkeep*: this function is called by the *Automation Nodes*, looking for "*upkeepNeeded*" to return true;
- *performUpkeep*: this function is called by the *Automation Nodes*, after *checkUpkeep* returned true.

Using the flexibility of the *Chainlink Keepers* system, a single Smart Contract can be registered to be checked by the *Automation Nodes* for distinct sets of conditions, allowing to build different paths. When a set of conditions for a path is true, the *Automation Nodes* will call the corresponding *performUpkeep function*, in accordance to this specific path.

Thus, the Lottery Contract makes use of 3 *Custom logic trigger*s, checking for 3 different set of conditions, to trigger 3 different paths.

These 3 paths are detailed in the following sections.

1.2.4. Picking the Winner

After the players entered the game and while the Lottery is running, the *Chainlink Automation Nodes* are responsible for checking if the time to play has passed. If this is the case, the *Automation Nodes* send a request to the *Chainlink VRF* to get a random number, which will be used to pick up the winner address, as described below.

1.2.4.1. Checking the end of play-time & Requesting a random number

The ***checkUpkeep path 01*** is needed to check when the Lottery play time has ended.

For the *checkUpkeep path 01* to return true, all the following conditions need to be true:

1. The Lottery state is *Open to Play*,
2. The Lottery time interval to play has passed,
3. The Lottery has at least one player, and hence the Lottery is funded,
4. The Chainlink subscription to the service has enough LINK (independent of the Lottery Contract state).

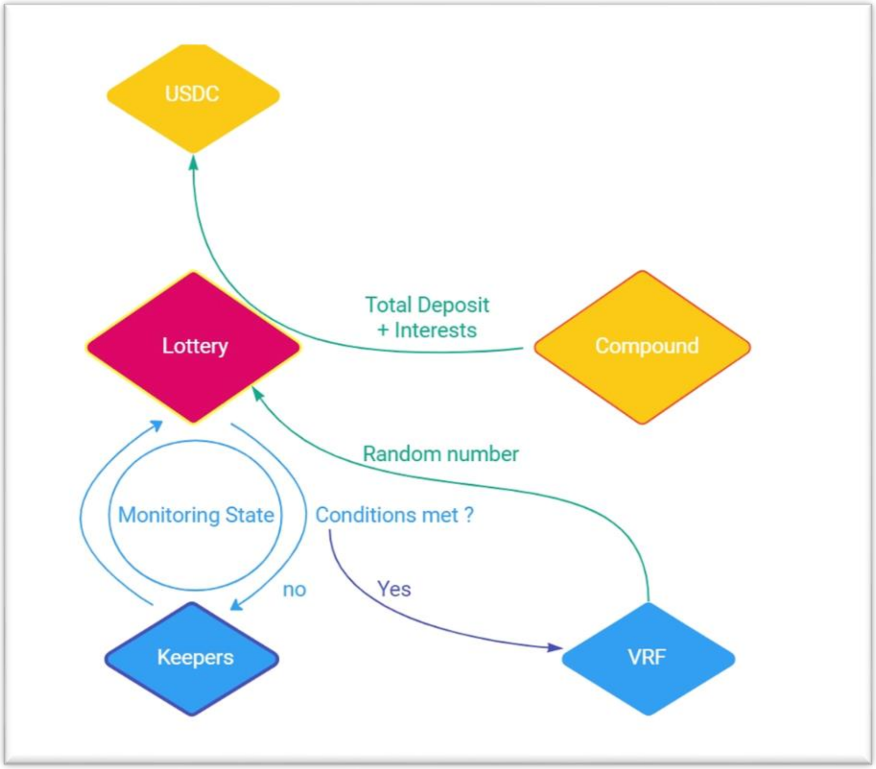


Figure 4: Chainlink Keepers – Triggering Chainlink VRF to get a random number

Thus, if the *checkUpkeep path 01* has returned *true*, the ***performUpkeep* function *path 01*** is called, triggering the following:

1. The Lottery contract calls the *Compound* contract ("*comet*") to transfer all available USDC to the Lottery,

2. A request for randomness is made to the *ChainLink VRF*,
3. The state variable *s_endPlayTime* is updated to the maximal uint256 value, to prevent the *checkUpKeep path 01* to return *true* before the next Lottery round,
4. The Lottery state switches from *Open to Play* to *Calculating Winner Address*.

1.2.4.2. Using the random number to select the Winner

When the request for randomness is resolving, the *Automation Nodes* call the *fulfillRandomWords* function of the Lottery contract, which triggers the following:

1. The Winner address is picked up from the current players, using the random number generated by the VRF,
2. The Lottery state switches from *Calculating Winner Address to Calculating Winner Gains.*

1.2.5. Calculating the Winner Gains

The ***checkUpkeep path 02*** described below is needed for two reasons: because a Smart Contract cannot by itself access to the data from the event logs, and because for the Lottery contract to be able to transfer the gains to the winner, we need to wait for the *withdrawal from Compound* transaction to reach finality (this transaction having been initiated by the *performUpkeep path 01*).

For the *checkUpkeep path 02* to return true, all the following conditions need to be true:

1. The Lottery state is in *Calculating Winner Gains*,
2. The Lottery USDC Balance is not null,
3. The Lottery USDC Balance on Compound is null.

When this *checkUpkeep* returns true, it means that the USDC withdrawal from *Compound* to the Lottery was successful, and the funds are now available from the Lottery contract. At this step, the Lottery contract is ready to calculate and to transfer the Gains to the Winner.

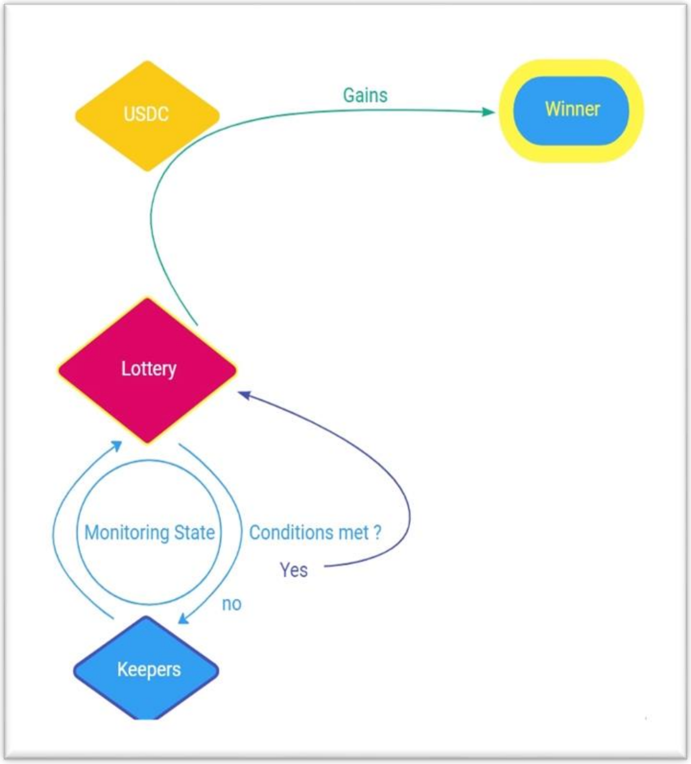


Figure 5: Chainlink Keepers – Triggering Gains Calculation & Transfer

Thus, if the *checkUpkeep path 02* has returned *true*, the ***performUpkeep* function *path 02*** is called, triggering the following:

1. The Lottery calculates and transfers the USDC Gains to the Winner,
2. The Winner address state variable and the Players array are reset to the *address(0),*
3. The Lottery state switches from *Calculating Winner Gains* to *Open to Withdraw.*

4. The state variable *endWithDrawTime* is set, by adding the current block timestamp value to the *i_intervalWithdraw* value. Thus, *endWithDrawTime* defines the time at which it will be no longer possible for players to withdraw their funds, for this round.

Without player action, by default, all players (including the winner) keep their USDC (all without gains) in the Lottery pool for the next runs, ready to be lend on the *Compound V3* protocol.

Also, all players (including the winner) hold their LTK tokens as long as they do not withdraw their USDC from the Lottery.

The Gains, won by the winner at the end of each Lottery round, are the interests accumulated by the Lottery pool on the *Compound V3* protocol, during each Lottery round.

1.2.6. Withdrawing from the Lottery

Following the previous step, any holder of LTK tokens (that is, the current players of this round, and the previous players still holding their LTK from previous rounds) can now withdraw its total USDC deposit, as the Lottery state is now in *Open to Withdraw,* and as long as the Lottery *time interval for the players to withdraw* their funds has not passed. The monitoring of these conditions is performed by the *Automation Nodes*, the next *performUpkeep* to be triggered being the *path 03*, described in the next section.

When a player calls the *withdrawFromLottery* function, all its USDC deposit is transferred to its address, and the player transfers the ownership of all its LTK token back to the Lottery Contract.

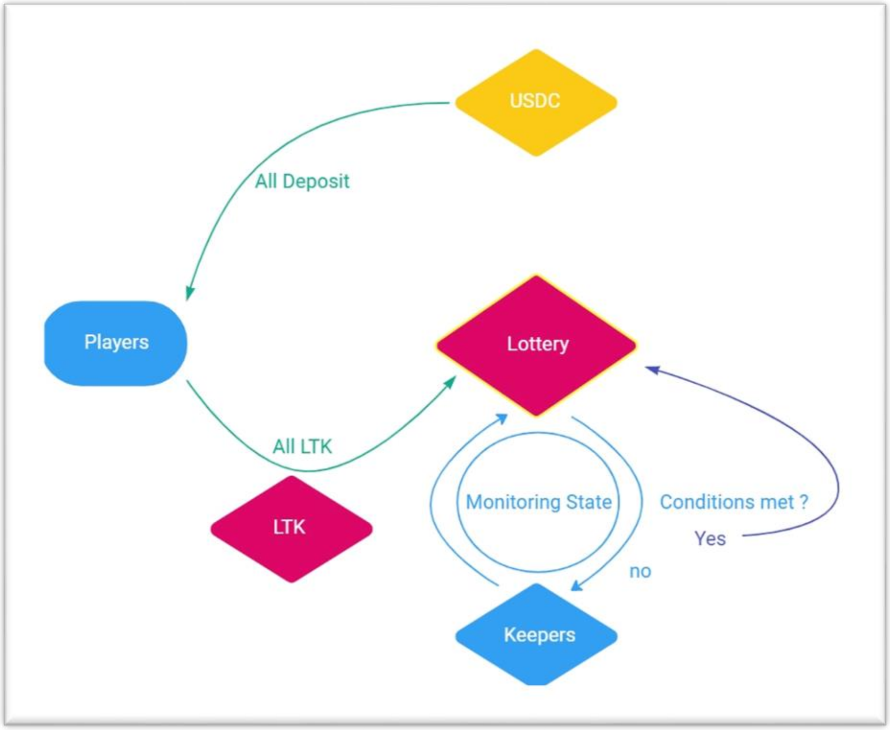


Figure 6: Chainlink Keepers – Triggering Withdrawal Time

1.2.7. Ending a Lottery round – Lending

The ***checkUpkeep path 03*** is needed to check when the time for the players to withdraw their funds has ended, for the current Lottery round.

For the *checkUpkeep path 03* to return true, all the following conditions need to be true:

1. The Lottery state is in *Open to Withdraw*,
2. The Lottery Time interval for the Players to withdraw their funds has passed.

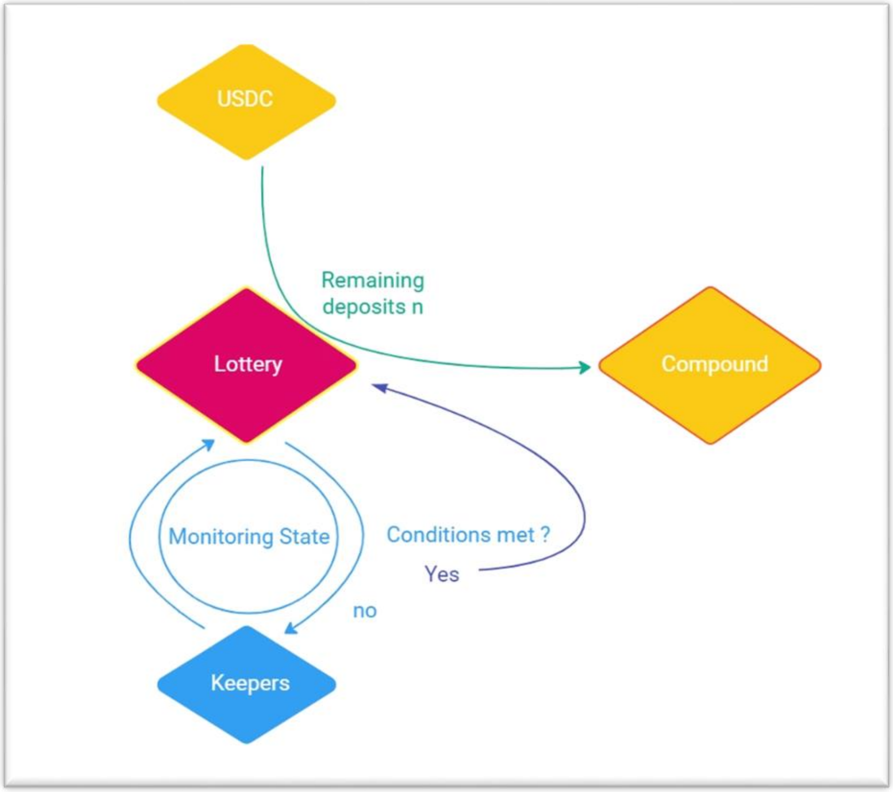


Figure 7: Chainlink Keepers – Triggering Lending

Thus, if the *checkUpkeep path 03* has returned *true*, the ***performUpkeep* function *path 03*** is called, triggering the following:

1. The Lottery state switches from *Open to Withdraw* to *Open to Play*,
2. The Lottery Contract supplies *Compound* with all current Lottery USDC balance, then starting again to generate interests.

At this stage, a new Lottery round is ready to start, which will be triggered by the next First Player entering the game.

1.2.8. Admin functions

The Owner of the Lottery Contract, defined during the contract deployment, has access to admin specific functions, allowing him to:

1. Transfer the ETH Lottery balance to its own address,
2. Transfer USDC from its address to the Lottery Contract, and supplying *Compound* with this USDC amount.

II. Business Case

In this chapter we will discuss why a *no loss* decentralized lottery matters, and how it differentiates from and revolutionize the traditional lottery systems.

II.1. Problem

Below are the problems our Lottery DApp is addressing.

II.1.1. Central authority means single point of failure

Traditional lotteries require the players to trust the central authority running the lottery as actually running it honestly. Unfortunately, this is not always the case.

- **cheating on randomness:** as an example, among others incidents originating from this single point of failure, in 2015 a lottery was rigged[xiii] for over $14 Million by a lottery security official, who fraudulently controlled, towards its sole benefits, the outcome of the random number generator used by the lottery.
- **lottery rug pulls:** another problem with traditional lotteries is that, laws and regulations apart, a central authority can technically decide to unilaterally break its contract with its customers while a lottery is running, after the tickets have been sold, and then simply run away in a kind of lottery rug pull.

II.1.2. Traditional lotteries, a *one winner* and *all other losers* game model

In traditional lotteries, the authority running the lottery is usually the only one who always win. Indeed, among all players, only one (or few ones, depending on the lottery set up) wins the gains, while all other players lose the investment they made by buying their ticket. It is noteworthy to mention that no value is generated from traditional lottery systems, apart from the ticket sales.

II.1.3. Traditional lotteries and the transient lottery ticket value

In traditional lotteries, one ticket is valid only for a single lottery round. For a chance to win a lottery round, a player must take the risk of losing its investment at every single round. Moreover, a traditional lottery ticket only holds the potential value corresponding to a (transient) chance to win.

II.1.4. Traditional lotteries, centralized governance model and passive players

In traditional lottery systems, the governance is fully centralized, and the players are passive participants of the game. Their interaction with the game is only about buying a ticket for a chance to win. And this interaction is of short time, as it lasts only for a lottery round.

II.2. DApp/Solution

Below are the solutions offered by our Lottery DApp to address the listed problems above.

II.2.1 Decentralization for a provably truly randomness and incorruptible lottery

We believe that a player should never have to worry that the people running the lottery have the ability to cheat. Indeed, an essential part of any sustainable lottery is the trust that each entrant has in their money being treated fairly.

- **decentralized true random number generators (RNG):** decentralized lotteries built around blockchain technologies can fix this problem by providing a platform where the numbers chosen are truly random; moreover, by using cryptographic proofs, everyone can check that these numbers are truly random. The *Chainlink VRF* oracle, used in our Lottery, is a renowned standard for decentralized RNG.
- **decentralized trustless automation:** using smart contracts living on a blockchain, immutable lottery rules, like the given *time to play*, the *time to withdraw*, can be programmatically enforced to be executed. Moreover, the variables state of a smart contract can be monitored by decentralized automation systems such as the *Chainlink Keepers – Automation, to trigger* on-chain events in a programmatic and tamper-resistant way.

The immutable Lottery rules (written in the Smart Contract code), combined with the decentralized oracle services to generate true randomness, provide our players with a trustworthy experience.

II.2.2. Decentralized no-loss lotteries, integrating DeFi products, a game changer in the gaming world

DeFi protocols such as *Compound,* allowing participants to lend crypto assets to generate interests, constitute an interesting source of value, in order to generate the gains awarded to the winners. Moreover, unlike traditional finance, anyone can interact with the DeFi protocols in a permissionless way; anyone (or any smart contract) with a crypto wallet can access DeFi protocols built on *Ethereum*.

As opposed to the isolated business model of a traditional lottery, our decentralized lottery, built on a permissionless blockchain, can interact with DeFi protocols, and as such can contribute to the overall value generated by the whole ecosystem.

By lending the value generated by the ticket sales, and thus by providing liquidity to *Compound* DeFi protocols, our decentralized lottery is generating value from the interests earned on lending, apart from the ticket sales. The interests produced by lending are used to reward the winner for each lottery round.

II.2.3. Lottery, perpetual chance to win, and ticket value beyond the chance to win

As opposed to traditional lotteries, our Lottery allows players to invest in a lottery ticket and to keep their chance to win, for all lottery rounds that will happen thereafter. Indeed, as long as the players keep their USDC deposits in the Lottery pool, they keep their chance to win, without having to reinvest in a new ticket at every lottery round. This creates a win-to-win situation, for every single player, keeping their chance to win over a potentially infinite number of lottery rounds, and also for all players as a community, as the more value is deposited and locked into the lottery pool, the higher the yield and the gains will be for the winners.

Besides this perpetual chance to win offered to the players, the fact that a Lottery Token (ERC20) is offered to each player buying a lottery ticket, also constitutes another source of value for the players, when is considered the potential appreciation over time of the LTK tokens.

II.2.4. Decentralized lotteries and the power of DAOs

It is anticipated that our Lottery project will, in future implementations, integrate the features of a DAO. A decentralized lottery built around as DAO offers financial and social advantages, as compared to traditional lotteries.

The DAO will give to the *LTK Governance token* holders the possibility to participate in the governance decision-making process of the Lottery project. For instance, these decisions could range from the price of the lottery entrance fee, the DeFi products used to generate the gains, or the calculation of the gains' distribution.

Besides the financial aspects of these decisions, the DAO allows to build a social engagement around the project, with players building together towards shared goals. We anticipate that the community built around the Lottery project will attract more players in the game, thus overall positively driving the project growth, where the social and financial aspects are closely intermingled.

II.3. Unique Value Proposition

Winning by Saving.

Players can be confident that the stated Lottery rules will always be the final rules, due to the trustless and tamper-resistant environment of the Ethereum blockchain. These rules enforce execution of a truly random chance-based game.

Players are incited to lock their savings, to get a chance to win gains without taking any risk, they can win but cannot never lose.

II.4. Customer Segments

Our Lottery will create value for the following customer segments:

- Usual chance-based game players looking for more fair lottery games, where they can win but cannot never lose;
- Investors willing to diversify their portfolios, speculating for the Lottery LTK token to appreciate over time;
- Anyone willing to take a chance to earn gains on its savings;
- Online casinos and gaming platforms willing to offer innovative gaming experience to their customers.

II.5. Unfair advantage

We believe in the support of the GBI community to help us spread the word.

II.6. Channels

The value will be transported to our customer segments through:

- *Usual chance-based game players* customer segment: advertisement of educational content related to gaming on social media;
- *Investors* customer segment: advertisement of educational content related to investment and savings on social media;
- *Online casinos and gaming platforms* customer segment: referrals pointing to our lottery platform;
- Players' cross-friends recruitment. As the more the lottery pool is funded, the higher are the potential gains, we believe in an exponential growth in the players-to-players recruitment.

II.7. Key Metrics

The success of our Lottery will be evaluated by the evolution over time of the number of Lottery Tokens hold by players, and eventually by the appreciation of the LTK token value over time.

II.8. Cost Structure

The implementation of our Lottery will include the costs associated with:

- Developers and Designers,
- Contracts deployment on Ethereum,
- Chainlink Oracles base subscription and operating costs,
- Marketing,
- Lawyers for Lottery compliance to laws and regulation.

II.9. Revenue Streams

The revenue streams for our Lottery are constituted of:

- ETH from the sales of the lottery tickets,
- LTK token appreciation over time,
- Affiliation links to other gaming platforms,
- Sales of Lottery branded physical products,
- A la carte Lottery DApps development for gaming platforms.

Conclusion

The no loss decentralized Lottery system described in this paper clearly disrupts the traditional lotteries scheme. These no loss lotteries already exist (*PoolToGether*[xiv]), but I wanted to take the opportunity of my Master Thesis to dive deep into these systems and sharpen my coding skills on the *Ethereum* blockchain.

Thanks to blockchain technologies, by providing players with a trustless gaming environment, players do not have to worry anymore about the potential dishonest behavior of a central lottery organization. Players just need to trust an immutable code and its deterministic execution, which is verifiable by all, because deployed on the public *Ethereum* blockchain.

Nonetheless, because of their deterministic behavior, blockchain systems cannot generate true random numbers. Thus, for our Lottery to be a pure chance-based game, the true random number generation is performed through the use of the *Chainlink VRF* oracle service. This consensus-based oracle network constitutes a tamper-proof and verifiable source for random number generation, by using two keys to generate a random but unpredictable value that can be verified through proof of correctness.

As opposed to traditional lotteries gains distribution scheme, where only one (or a few) player wins all gains and all other lose their investment, a no loss lottery game is a game changer, by not only rewarding a winner but also allowing all other players to preserve their funds. This is made possible with a decentralized Lottery system, by interaction with permissionless DeFi protocols, such as *Compound*, to lend the players deposits in order to generate interests, which finally constitutes the gains of the winner.

Another interesting aspect of our Lottery is built around the idea that, as opposed to traditional lotteries, once a player buys a ticket, the life span of this ticket is by default, forever. Indeed, as long as a player keeps its funds in the lottery pool, thus participating in the lottery value creation by contributing to the lending pool, this player in return enjoys an infinite chance to win, as his ticket has the potential to be picked as the winner one, for every lottery round that will happen next.

Moreover, as for each lottery ticket bought, a Lottery Token is transferred to the player, the lottery ticket now holds two promises: an (infinite) chance to win the lottery gains, and, a chance to see the value of the Lottery Token to appreciate over time.

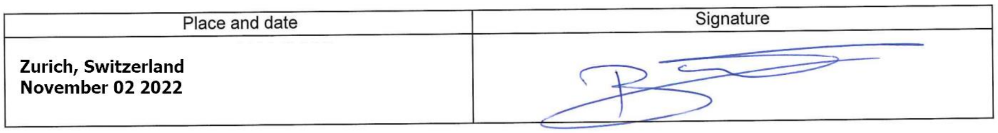
Another interesting set of features, yet to be implemented in our current Lottery system, wraps around the concept of a Decentralized Autonomous Organization, which will offer financial and social advantages, as compared to traditional lotteries. Indeed, the Lottery DAO will give the LTK token holders the possibility to participate in the governance decision-making process of the Lottery project. This is anticipated to build traction to the Lottery, by inciting players to buy more tickets and stay in the Lottery, as offering them the opportunity to give their voice and actively participate in its evolution, with the power of each player's voice being a function of the number of LTK tokens hold by this player.

Besides the possibility, offered to LTK token holders, to participate in the financial decisions of the Lottery (lottery ticket price, lottery gains distribution scheme etc...), the DAO allows to build a social engagement around the project, with players building together towards shared goals. We thus anticipate that the community built around the DAO Lottery project will attract more players in the game, positively impacting the project growth.

Finally, compared to traditional lottery games, our no loss decentralized Lottery dramatically enlarge the scope of customer segments. First, as a *winning and never losing* game, we anticipate that the pool of traditional chance-based game players will be enlarged by a category of new players, a priori reluctant to the financial risk associated with chance-based games, now reassured by the fact that they cannot lose their investment. Secondly, for the same reason described above, but also in conjunction with the fact that the Lottery Token might see its value appreciate over time, a new customer segment made of investors are expected to be attracted to our Lottery. Finally, we believe that online casinos and other gaming platforms willing to offer new gaming experiences for their clients might constitute another customer segment.

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree.



Place and date	Signature
Zurich, Switzerland **November 02 2022**	

References

[i] Ethereum blockchain: Link
[ii] Solidity, an object-oriented programming language for implementing smart contracts on Ethereum : Link
[iii] Goerli Test network: Link
[iv] Polygon Layer-2 blockchain: Link
[v] OpenZeppelin Smart Contracts Library: Link
[vi] “How to Generate Truly Random Numbers in Solidity and Blockchain”, *Better Programming* article: Link
[vii] Chainlink VRF oracle (RNG): Link
[viii] Compound V3, EVM compatible DeFi protocol: Link
[ix] USDC ERC20 token: Link
[x] HardHat, a development environment for Ethereum software: Link
[xi] Ethers.js, a library for interacting with the Ethereum Blockchain and its ecosystem: Link
[xii] Chainlink Keepers-Automation oracle: Link
[xiii] Rigged Lottery, *New York Post* article: Link
[xiv] PoolToGether: Link