Decentralized Blockchain Betting System

Aliaksei Matsarski*, Neha Yawalkar*, Priyanka Subramanyam*, Shivani Mangal*

San Jose State University

Abstract—Lotteries are one of the most popular forms of betting. Today's gambling industry have a bad reputation because most of the odds are in favor of the owners, which makes the system unfair for the users. The decentralized blockchain betting system address this problem by having a system which is fair and transparent without the involvement of a third party. The system uses Ethereum blockchain which is an open source software development platform which enables the developers to deploy decentralized applications. Astrum blockchain betting system is based on the Ethereum blockchain, where users can play against each other without a third-party involvement.

Index Terms—Ethereum blockchain, Lottery, Distributed System, Secure transactions, Authentication.

I. INTRODUCTION

Currently with the growing usage of mobile phones and internet, online lotteries are played everywhere. Until now most of the online lottery application were based on business to consumer model. With the growing technology there are many companies offering a lottery platform but ensuring a fair game remain as a biggest challenge in lottery. The blockchain lottery can solve this problem related to the trust and fairness of the game. In today's world the lottery industry is moving more proximate to high technology lotteries, and decentralized blockchain lottery is an innovative step towards this technological advancement.

Blockchain: A blockchain is a decentralized, public ledger of all cryptocurrency transactions. A block is the 'current' part of a blockchain, which records some or all the recent transactions. Once completed, a block goes into the blockchain as a permanent database. Each time a block gets completed, a new one is generated. There is a countless number of such blocks in the blockchain, connected to each other. [2]

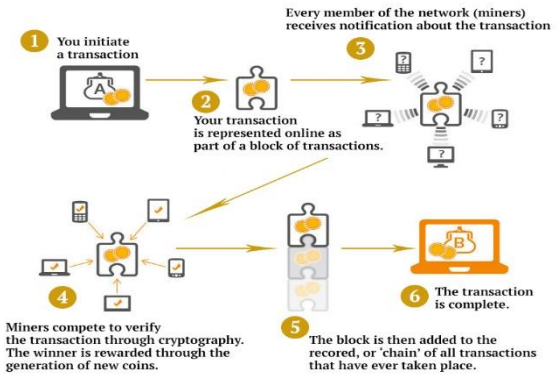

Fig 1. Blockchain Transaction

II. DESIGN OBJECTIVES

The main objective of this paper is to create a **distributed lottery system**, where players could bet on a specific outcome to win 'Ethers'. Some of the other goals include:-

1. Creating **smart contracts using Solidity** to hold various functionalities for the players and admin, that can be displayed on the front end.

*All are first authors of this article.
Written in the format of the IEEE Computer Society

2. **Authenticate** players login using third-party **Metamask** authentication
3. **Automated Testing** of contracts using the **Truffle web** framework.
4. Usage of **Private** Ethereum blockchain using **Ganache** within Client-side script to import **private keys** for players.
5. **Client-side scripting** to create front-end with nice 'look and feel' using **AngularJS**.
6. Effectively manage **workflow from smart contracts to front-end** with the usage of **web3 API**.
7. **Server-side web programming** to design a **separate path for the admin** to control the game, view statistics of games along with users using **Node Js, Express Js**.
8. Authenticate and **secure admin** login using **Passport JS**.
9. Creating **test cases** for every pitfall that could occur **within the code** and managing **borderline situations** effectively.
10. Deployment of **docker** containers of web application and blockchain on an **EC2 instance**.

III. ARCHITECTURE

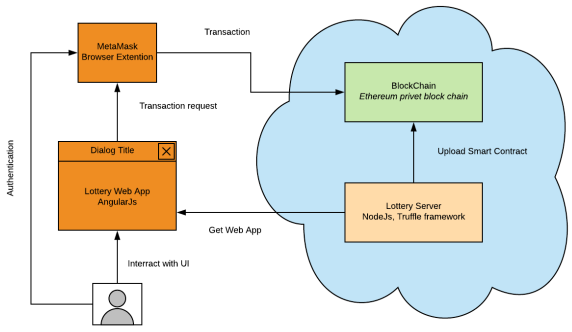

Fig 2. Application Architecture

IV. IMPLEMENTATION

For implementation, an agile development approach took place and it was accomplished in two sprints. During the first sprint, all the required environments were established:

i. AWS es2 instances;
ii. Docker account to store Docker containers;
iii. Bamboo server to perform code building;
iv. an Atlassian account for a) running Jira to accommodate Project management purpose and b) exploiting Confluence to organize documentation;
v. Ethereum private blockchain deployed on AWS ec2 instance (t2micro).

During the second sprint, the main functionality of the project was implemented. The cornerstone of the project is Smart Contract (SC) - a standalone small 'application' deployed on the blockchain and containing the business logic. The Smart Contracts are written by using the Solidity language. To manage SC in blockchain, a Truffle framework was involved; while for the web server, an Express NodeJs framework was used. Furthermore, on the client-side Bootstrap and AngularJs took place.

In addition, MetaMask - a web browser extension - served as a proxy for the communication between web application Javascript and Blockchain. Being responsible for user authentication and transactions performance, MetaMask is the one calling blockchain. This way, it is not necessary to store users' passwords and wallets as all the authentication part occurs in the MetaMask app.

Finally, Docker container was used for a deployment process of the web server. The web server, in turn, was deployed on the same ec2 instance as the blockchain. Though not

necessary, that step was performed for the resources efficiency reason.

V. CONCLUSION

Astrum is a decentralized betting platform which offers a secure way for transactions to the users and is unbiased due to lack of a central authority. It currently supports one game but has the potential to be a platform where users can play multiple games at any point of time. Casinos can run Astrum on their private blockchain, thus offering the users an experience of the play without having to be present at the casino. Astrum is built on Ethereum framework and thus follows the proof-of-concept algorithm to build consensus among all its nodes. Thus, Astrum explores the possibilities that blockchain has to offer with an elegant UI, built on Node Js to support it.

ACKNOWLEDGEMENT

Knowledge is an experience gained in life. It is a possession which should not be shelved but should be happily shared with others. In this regard, we are extremely thankful for Professor Rakesh Ranjan's valuable feedback and direction for the completion of this project. We would like to express our gratitude and thanks to professor, for his encouragement and guidance.

REFERENCES

[1] https://blockgeeks.com/guides/ethereum/
[2] https://www.investopedia.com/terms/b/blockchain.asp
[3] https://blog.sagipl.com/blockchain-development/
[4] https://digitalgoldclub.com/2017/12/12/lesson-3-how-does-cryptocurrency-work/