DSE: Decentralised Stock Exchange Using Blockchain

Team 10: KATtana
Kushagra Agarwal[1], Akshit Garg[1], Tathagatha Raha[1]

[1] International Institute of Information Technology, Hyderabad, India

I. INTRODUCTION

Consider a world in which you could purchase stocks, bonds, derivatives, and cryptocurrencies from anywhere in the globe, 24 hours a day, seven days a week. Trades take place directly between two investors on this exchange rather than through a complicated latticework of brokers, clearinghouses, and other intermediaries and gatekeepers. Instead of lasting up to two days, they settle or close practically instantaneously. The system is less expensive, more transparent, and appears to be more open. Here in this report, we will try to create this system, which we call DSE, a decentralised stock exchange using blockchain.

Stock exchanges provide a safe and regulated environment in which market participants can confidently trade shares and other qualified financial instruments with zero to low operational risk. They function as both primary and secondary markets, according to the guidelines established by the government. NSE (National Stock Exchange) and BSE (Bombay Stock exchange) for example are the largest exchanges by volume in India.

The stock market is one of the most vital components of a free-market economy. It allows companies to raise money by offering stock shares and corporate bonds. It lets common investors participate in the financial achievements of the companies, make profits through capital gains, and earn money through dividends. The stock market works as a platform through which savings and investments of individuals are channelized into the productive investment proposals which in the long term helps in capital formation and economic growth for the country.

The New York Stock Exchange is the worlds largest stock exchange with a market cap of $26 trillion! BSE in India has a market cap of Rs 255 trillion and NSE has a cap of Rs 89 trillion. The average daily trading volume on the NYSE alone spans between 2 billion and 6 billion shares. Imagine the amount of transactions happening daily based on trust on centralized for-profit institutions! This alone is motivation enough to remedy the pitfalls of the current exchange system and try and design the most efficient one.

This study does precisely that; we try to incorporate the advantages embraced by blockchains to do away with the problems of the currently inplace stock exchange markets. The direct beneficiary of such a remedy is the trader (you and me) who currently pays a hefty transaction fees over each transaction. The goal is to DISTRIBUTE this TRUST on a centralized agency using BLOCKCHAIN as our yardstick.

II. Why do we need Blockchain?

A. Current Centralised Stock Exchange System

As shown in Figure 1, the typical stock exchange market system is designed as a centralised system in which one component collects all market activities from the agents. These activities are stored in the Security Exchange Platform's central register. They are classified as bids or offers and are sorted by price. The matching is then done at discrete points in time, based on the actual buy-sell orders.

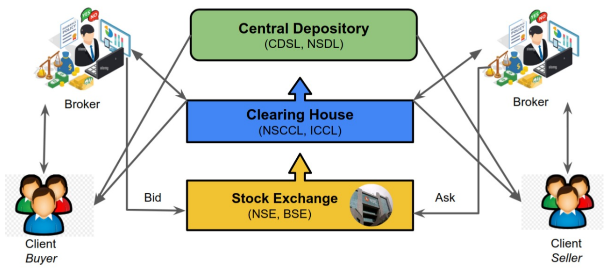


Fig. 1. Centralised Architecture of Stock Markets

Once market activities are recorded in the Security Exchange Platform, they are carried out in accordance with their terms. Following the matching of the acts, the actual transfer must be carried out, which implies that the real asset (bond, share certificate, etc.) must be swapped for the actual monetary value. To ensure that the transfer goes smoothly, the Clearing House is responsible for performing all of the essential stages, such as guaranteeing and documenting all transactions and categorising operations by member for each trading day. The Settlement Platform guarantees that the items are really exchanged. In the current systems, stock securities are settled in 2 business days (T+2), while government bonds are settled in one day (T+1).

B. *Problems with the Current System*

Despite its widespread acceptance and adoption, the traditional stock market platform design has a number of limitations and flaws, including the following:

- Large transaction fees and monopoly of central authority on this levied fees.
- Lack of transparency in the process for the trader.
- Because of centralised architecture, each participating system, such as brokers and the stock exchange, is considered a single point of failure.
- When a system fails, the data it manages is inconsistent, resulting in mistakes and a longer recovery time.
- The long time the system takes to perform financial settlements. Indian Exchanges like NSE follow a T+2 days for delivery protocol for trades done in the equity market.

C. *How does Blockchain solve these Problems?*

Interoperability, trust, and transparency difficulties in fragmented market systems may be addressed via blockchain. Through automation and decentralisation, blockchain can improve the efficiency of stock exchanges. It has the potential to help clients save a lot of money on commissions to brokers while also speeding up the transaction settlement process. It will be useful in securely automating the post-trade process, reducing trade paperwork, and transferring legal ownership of securities. The laws and regulations would be in-built inside smart contracts and enforced with each trade in order to register transactions, with the blockchain network functioning as a regulator for all transactions, removing the need for third-party regulators to a great extent.

1) Lower Transaction Costs: Transactions on the blockchain are quicker because trade confirmations are done by peers using smart contracts rather than an intermediary. As the number of intermediaries in the system decreases, expenses associated with them, such as trade record keeping, audits, and trade verifications, decrease. A small gas fees would still be charged which is much smaller compared to transaction fees for large dealings.

2) Fairness and transparency mechanisms: Because the blockchain ledger is built so that all nodes have access to the whole record of transactions and, as a result, investor holdings, it may offer perfect transparency and confidence to the market.

3) Robustness: The inherent advantage of a distributed database maintained in a peer-to-peer network (P2P) is that each honest node has a copy of the current state of the blockchain. Therefore, the whole network as such is majorly robust and can never be corrupted as long as the majority of nodes are honest.

4) Faster post-trade events: Post-trade operations using blockchain and smart contracts can eliminate the need for intermediaries, minimise counter-parties and operational risk, and provide the framework for speedier transaction settlement. Financial institutions may settle securities in minutes rather than days, resulting in simplified real-time settlement, enhanced liquidity, supply chain efficiency, and increased transparency.

III. Problem Statement

We propose a novel blockchain-based architecture for a fully decentralized stock market. Our architecture is based on a permissionless Ethereum blockchain wherein smart contracts would encompass the business logic of the day-to-day trades in the market. The potential of the blockchain system can bring benefits to the entire system, by leveraging on full replication and verification by all the nodes in the system regarding the assets owned, the execution of the market orders and the correct settlement between the accounts. Furthermore, the ensured immutability of the ledger brings a valuable advantage with respect to the centralized system. Also, by decentralizing the system, there is no central authority or intermediate needed for placing and executing the orders. Since this new platform does not introduce significant changes to the stock exchange trading logic and does not eliminate any of the traditional parties from the system, our proposal promotes efficient adoption and deployment of decentralized stock exchange platforms.

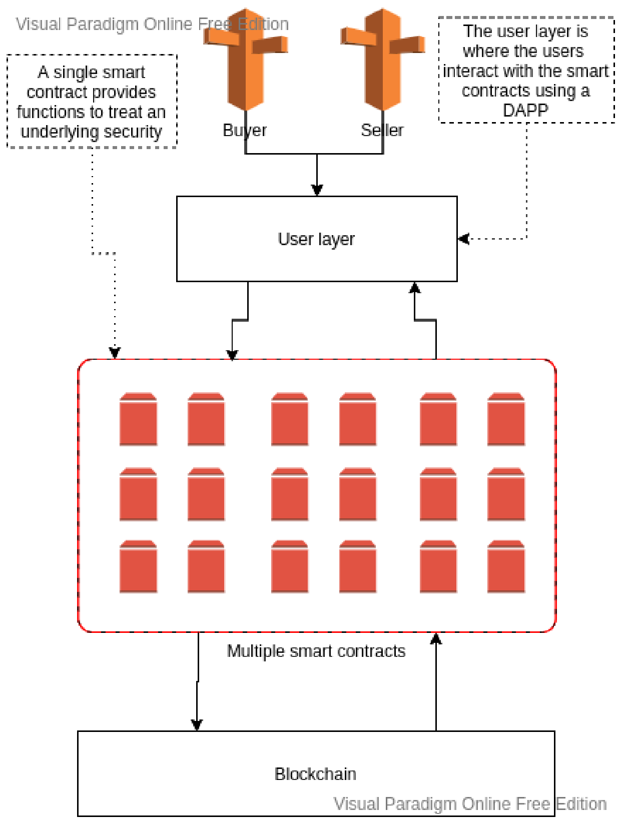
IV. Proposed Solution

We propose an Ethereum based solution to implement a decentralised stock exchange which will try to solve the problems with existing centralised exchanges. The detailed technical solution is mentioned in the sections below.

A. Architecture

This section details the architecture of the system proposed for the DSE. The system will consist of three main layers, namely the user layer, smart contract and the blockchain the architecture can be understood using the Figure 2.

1) User Layer: This is a web based front end with which user will interact to analyse the market and post buy and sell orders accordingly. The front end will interact with multiple smart contracts to simulate the behaviour of a Stock Exchange. Each smart contract will be responsible for execution and storage of trades of a single security.
2) Smart Contracts: Each smart contract will execute trades and store orders of its underlying security. Hence, the system will have multiple smart contracts each representing different underlying security.
3) Blockchain layer: We will be using the existing architecture of Ethereum blockchain for execution and validation of the smart contracts.



Visual Paradigm Online Free Edition
A single smart contract provides functions to treat an underlying security
Buyer
Seller
The user layer is where the users interact with the smart contracts using a DAPP
User layer
Multiple smart contracts
Blockchain
Visual Paradigm Online Free Edition

Fig. 2. Architecture of the proposed system

B. Smart Contract

This subsection details about the smart contract which would be used to process transactions and store orders for the underlying security.

State variables of the Smart Contract:

1) Symbol - A string, denoting the name of the underlying security to be transacted by the smart contract *example TATA Steel*

2) OwnedStocks – A mapping to store the owner address and the quantity of the assets owned
3) PlacedOrders - A uint, representing number of orders placed so far for the given symbol
4) MarketPrice – A uint, representing the price at which the latest action was executed
5) SellOrders – An array of type NewOrder Structure of all the selling orders which are yet to be executed.
6) BuyOrders – An array of type NewOrder Structure of all the selling orders which are yet to be executed.

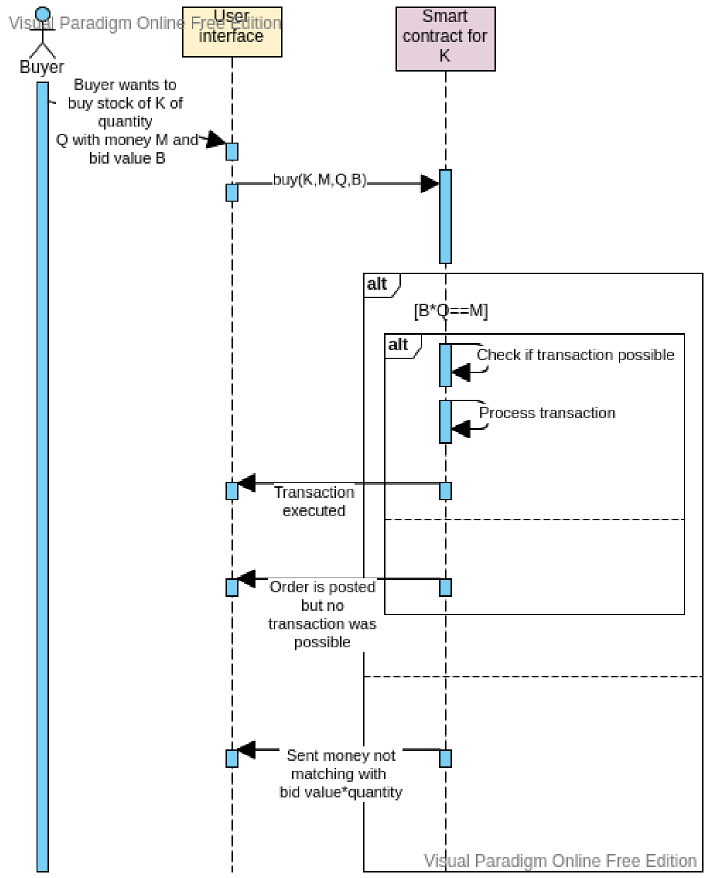
Structure of the NewOrder struct:

Each Order action is defined by the smart contract as a structure named NewOrder with the following variables:

1) OrderID - uint, representing the unique order id
2) State - A uint representing if order is open for execution(0), completed(1) or cancelled (2).
3) ExecutorAddress - A payable address, representing the address of the individual who posted the order.
4) PostingTime - A string, which is the timestamp of the form yyyy-MM-dd HH:mm:ss.SSS when the action was posted to the smart contract.
5) CompletionTime - A String, which is the timestamp of the form yyyy-MM-dd HH:mm:ss.SSS when the transaction was completed by the smart contract.
6) SellOrBuy - A String, representing if order is buy or sell order.
7) Price - A uint, representing the price investor is willing to give or take for the order.
8) OrderType - A String,representing if order is StopLoss or Normal order
9) LossLimit - A uint, representing price at which stop loss needs to be triggered,
10) Availability - A string, representing if it is an Open / Immediate-or-Cancel (IOC) order.

Functions of the SmartContract:

1) Buy - A function to post a buy order for underlying security, function will take all the necessary inputs to post a buy order as arguments and will return the status of order. The function will is a payable public function and when an investor places a buy order the amount of money required for complete execution of the order will be paid upfront to the smart contract by the buyer. The Smart contract will verify the amount of money paid is satisfactory, if so it will place the buy order. Upon addition of a new buy order smart contract will check for any possible transaction and will complete the transaction if any. Figure 3 can be referred to understand the sequence of the function.
2) Sell - A function to post a sell order for underlying security, function will take all the necessary inputs to post a sell order as arguments and will return the status of order. The function will check if the address posting the sell order owns the quantity of



Visual Paradigm Online Free Edition
Buyer
User interface
Smart contract for K
Buyer wants to buy stock of K of quantity Q with money M and bid value B
buy(K,M,Q,B)
alt
[B*Q==M]
alt
Check if transaction possible
Process transaction
Transaction executed
Order is posted but no transaction was possible
Sent money not matching with bid value*quantity
Visual Paradigm Online Free Edition

Fig. 3. UML sequence diagram representing the buy functionality

security to be sold, if so the contract will move the amount of quantity to be sold from sellers address to contracts address. The function will then add the sell order and execute the possible transactions if any.

3) CancelOrder - A function to cancel an existing order. Function will take order ID of the order to be cancelled as argument, it will then check if the person cancelling order is also the one who posted it. If so the contract will cancel the order. If the cancelled order is of type buy contract will send back the money allocated for buying the security while in the case of sell order the security which was secured by contract or selling will be sent back to seller.
4) getOrderStatus - A function to get state of order, function will take all the necessary inputs to verify state of order like order id and will return state of order. The function will also verify if the address asking the details of the order is the same who posted the order.
5) getMarketPrice - A function to get last transaction price of the underlying security.
6) getMarketDepth - A function to get the market depth of the underlying security.

C. What blockchain form will be suitable?

The Proof-of-Stake (PoS) form of blockchain will be the most suitable for the above application. PoS provides a permissionless network, with extremely low energy consumption compared to its predecessor blockchains of the Proof-of-Work (PoW) form. PoS also provide higher throughput with a much lower computational needs if compared with the PoW blockchains. PoS also discourages any attack from the miners as miners themselves hold a stake in the blockchain. We plan to develop a decentralised stock exchange using the Ethereum blockchain which is the world's current largest PoS blockchain. Ethereum runs the smart contracts written in solidity which is an easy to understand Object Oriented Programming Language.

V. Analysis

While our decentralised stock exchange has a number of advantages, the one major drawback would be scalability. As mentioned previously, the number of trades happening per day around the world is of the order of 10^9. Handling such a huge amount of data would make our blockchain very big and redundant due to increased latency. One possible solution to scale the system efficiently would be to store the hash of a transaction on the blockchain instead of the transaction itself. This would reduce space and time complexity of the scripts without compromising privacy and security. The mapping of these hashes to the original transaction and the original transaction itself would have to be stored separately in a shared database.

Another one of Ethereum's major challenges is the high gas fees. In Ethereum the gas fees is paid using the native currency "ETH" (ether). In the conventional stock exchanges, the broker gets a brokerage for each transaction and the broker in turn pays a fixed percentage (say 3%) of the transaction amount as transaction fees to the exchange. When the transactions are of high value, then this percentage fees scales linearly whereas, the gas fees is fixed (assumping constant space requirements for the contract). The gas fees is paid by the trader himself when he places an order to buy/sell. It is important to note that the gas fees might exceed the transaction fees for extremely low volume trades.

Newer blockchain technologies like Solana which use Proof-of-History (PoH) consensus algorithm could address the problems regarding transaction fees, efficiency, and scalability. By allowing validators to be in charge of their own clock, the transaction verification process is reduced since the nodes don't have to put in processing power before they can verify various timestamps. This improves the speed at which transactions are processed on the Solana network. The Solana network processes up to 60,000 transactions per second, surpassing that of Bitcoin, Visa, XRP and Ethereum combined. In addition to the transaction speed, the costs are significantly lower on the

Solana blockchain. As mentioned above, one of Ethereum's major challenges is its high gas fees. Users pay up to $50 to process a transaction on the Ethereum network. With Solana, the fees are significantly lower, usually around $0.00025 per transaction. Overall Solana is considered to be an Ethereum killer because of its innovation and how it is tackling most of Ethereum's weaknesses. Thus replacing Ethereum with Solana for example could help mitigate some of the impediments.

VI. Conclusion

Blockchain has the potential to disrupt financial services, especially in terms of automating market monitoring and post-trade event processing. It can directly address problems relating to data loss, data fragmentation, insider trading, and excess intermediaries. But this new technology can also give rise to difficult legal and regulatory challenges brought along with the risks of maintaining security standards across a decentralised database, and concerns around scalability. One simple solution to solve the problem of scalability could be to only store the hash of a transaction in the blockchain and store the mapping and the transaction in an external shared database.

Stock exchanges around the world are realising the potential of a decentralised exchange and are therefore trying to incorporate blockchain into their systems. In 2015, Nasdaq unveiled the use of its Nasdaq Linq blockchain ledger technology to successfully complete and record private securities transactions. The Securities and Exchange Board of India (SEBI) is also exploring how blockchain technology can be used in the Indian stock market. Recently, SEBI appointed an advisory committee, called the Committee on Financial and Regulatory Technologies (CFRT), for conducting research on the blockchain platform in the sphere of fundraising, asset management, and post-trade settlement.

Therefore, it is safe to assume that sooner or later, the monopoly of institutionalised for-profit exchanges will be shattered with the advent of a robust and efficient decentralised stock market powered by the disruptive market change Blockchain technology.

References

[1] Chen YH, Chen SH, Lin IC. Blockchain based smart contract for bidding system. In: 2018 IEEE International Conference on Applied System Invention (ICASI). IEEE; 2018. p. 208-11.

[2] Bhandarkar VV, Bhandarkar AA, Shiva A. Digital stocks using blockchain technology the possible future of stocks? International Journal of Management (IJM). 2019;10(3).

[3] Al-Shaibani H, Lasla N, Abdallah M. Consortium blockchain-based decentralized stock exchange platform. IEEE Access. 2020;8:123711-25.

[4] Noble D, Patil K. Blockchain in Stock Market Transformation: A Systematic Literature Review. REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS. 2021;11(4):5088-111.

[5] Miraz MH, Donald DC. Application of blockchain in booking and registration systems of securities exchanges. In: 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE). IEEE; 2018. p. 35-40.

[6] Lee L. New kids on the blockchain: How bitcoin's technology could reinvent the stock market. Hastings Bus LJ. 2015;12:81.

[7] Tapscott A, Tapscott D. How blockchain is changing finance. Harvard Business Review. 2017;1(9):2-5.

VII. Appendix

A. Pseudo-code for Smart Contract

```
// SPDX-License-Identifier: MIT
pragma solidity 0.5.16;

contract Symbol_DSE {
    struct NewOrder{
        uint OrderID; //represents the unique order id
        uint State; //represents id order is open for execution, cancelled or completed
        address ExecutorAddress;//represents the address of the individual who posted the
    order
        string PostingTime; //the timestamp of the form yyyy-MM-dd HH:mm:ss.SSS when the
    action was posted
        string CompletionTime; //the timestamp of the form yyyy-MM-dd HH:mm:ss.SSS when the
    action was completed
        string SellOrBuy; //represents if order is buy or sell order
        uint Price; //price individual is willing to give or take for the order
        string OrderType; // String, to represent if order StopLoss or Normal order
        uint LossLimit; // price at which stop loss needs to be executed
        string availability; // open or Immediate-or-cancel (IOC) order
    }

    string private Symbol;
    // string, denoting the name of the asset exchanged by the smart contract

    mapping (address=> uint) private OwnedStocks;
    // A mapping to store the owner address and the quantity of the assets owned

    uint private Placed_orders;
    // represents number of orders placed so far, for the underlying security

    NewOrder[] private Sell_orders;
    // NewOrder array, A sorted array of type NewOrder Structure of all the
    // selling orders which are yet to be executed

    NewOrder[] private Buy_orders;
    // NewOrder array, A sorted array of type NewOrder Structure of all the
    // buying orders which are yet to be executed.

    // A function to post a buy order for underlying security, function will take all the
    // necesarry inputs to post a buy order as parameter and will return the status of order.
    function buy() payable public returns(){
        // check if the value of money sent matches with quantity*price
        // place the order to array of buy orders
        // check if any transaction is possible
        // process all the possible transactions
        // return if order was executed or not
    }

    // A function to post a sell order for underlying security, function will take all the
    // necesarry inputs to post a sell order as parameter and will return the status of order
    .
    function sell() payable public returns(){
        // check if the adress posting order owns the quantity of security to be sold
        // place the order to array of sell orders
        // check if any transaction is possible
        // process all the possible transactions
        // return if order was executed or not
    }

    // A function to cancel an order for underlying security, function will take all the
    // necesarry inputs to cancel an order as parameter.
    function cancelOrder() public{
        // check if person cancelling order is also the one who posted it
        // check if order is not executed yet
        // if order not executed cancel order
    }

    // A function to get state of order, functiont will take all the necessary inputs to
    // verify state of order like order id and will return state of order
    function getOrderStatus() view public returns(string memory)
    {
        // check if person asking order status is also the one who posted it
        // return string representing if order is executed, cancelled, open.
    }

    // A function to get last trading price of underlying security
    function getMarketPrice() view public returns(uint)
    {
        // return last tranaction price of order
    }

    // A function to get the market depth of the underlying security
    function getMarketDepth() view public returns(string)
    {
        // return market depth of underlying security
    }
}
```