ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

EscrowChain: Leveraging Ethereum Blockchain as Escrow in Real Estate

Neel Kirit [1], Priya Sarkar [2]

B.E, Dept. of Information Science and Engineering, Bangalore Institute of Technology, India [1]

B.E, Dept. of Computer Science and Engineering, Bangalore Institute of Technology, India [2]

ABSTRACT: After the introduction of Bitcoin in 2008, cryptocurrencies and its underlying technology blockchain has seen tremendous growth. Other projects based on this platform have shown significant potential in areas beyond the transaction of currency between parties. While Bitcoin blockchain's focus is mainly on recording the ownership of digital money, Ethereum blockchain provides programming capability to be embedded into the coins. We try to exploit this feature to create a paperless escrow by utilizing the secure, open and distributed ledger model to keep a record of title history and enable faster transaction digitally.

In this paper, we introduce a virtual escrow model, the EscrowChain that uses smart contract to perform and verify the transfer of title ownership and money, the blockchain to keep a record of all transactions pertaining to a property while the decentralized network replaces the escrow agent or third party organization. We elaborate the design based on the Ethereum blockchain structure that can be used for buying and selling real estate. Further, we present an implementation of this model.

KEYWORDS: Blockchain, Ethereum, escrow for real estate, decentralized computing, digital title transfer, smart contract, distributed application, distributed ledger, consensus based transaction

I. INTRODUCTION

With the ever growing need of transferring information and money, it has become difficult for transacting parties to trust each other, thus creating the need of a reliable third party to manage and oversee the business. We do not lend money to others for this very reason and go to a bank which acts as the mediator. Similarly, any legal transaction from genesis to completion may involve a number of people and processes that can be time consuming. In real estate, buying and selling a property is a cumbersome task which can take up to several months; the stakeholders need to hire an escrow agent [14], verify title and past ownership history, register the property with a government registry office and so on. To help expedite the process and move the complete transaction online, blockchain [7] can be used.

Blockchain is a decentralized and distributed digital ledger [15] that contains records of transactions between parties. It is a global and transparent log that can be written by appending a block only, which makes any previous block immutable. The global ledger is secured using cryptography and can be verified by anyone on the network. The decentralized system ensures that no organization has complete control over the ledger making the transactions secure and tamper proof.

Ethereum [2] is based on blockchain that allows programming capability to be added to the ledger system. A contract can be coded to perform the transaction and the computers of the shared network can validate it. In a real estate transaction, this technology can be used to create a virtual escrow model: the title history present in the leger can be verified by the network, the program upon satisfaction of the conditions of sale can transfer the money and ownership to the respective parties and the purchase can be appended as a chronological block to the existing ledger. For the buyer, their purchasing capability can be verified instantly avoiding time-consuming visits to the solicitors and banks. For the seller, proof of home ownership including history of repairs, taxes, criminal records and liens can be authenticated. Thus, by removing the need of any third party interference and the promise of fraud-free transaction,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

Ethereum platform can be securely used to perform any type of transaction; a piece of code stored on the blockchain can simplify financial, legal and real estate transfers.

II. RELATED WORK

Bitcoin [1], introduced in 2008 and launched in 2009 is the first cryptocurrency that was decentralized, meaning no single party or authority has complete control over it. Anyone owning a bitcoin wallet can transfer money to another account which will be verified by the network. The participating nodes called miners [10] solve a mathematical problem to add a block the public ledger. Each block containing recent transaction is connected to the previous block using the cryptographic hash of the previous one thus forming an irreversible and immutable blockchain.

While Bitcoin allowed the transfer of digital currency between parties, the Coloured Coins project [3] launched in 2013 introduced metadata to the bitcoin. By adding color-aware transaction rules, a set of bitcoins having a specific color can be used to track and manage real world assets.

In 1997, work on smart contract by Szabo [4] showed that instructions can be embedded into hardware and software. With little to no dependence on human judgement, rules and agreements could be moved from paper and left to the algorithm to execute. Launched in 2013, Ethereum demonstrated how blockchain can be paired with a Turing-complete language. With the introduction of Ethereum, it became possible for any written logic (smart contract) to be run by the nodes in the network and update the ledger. It provides distributed data storage, computational capability and on successful creation of blocks, rewards the miners with it built-in cryptocurrency called Ether (ETH).

Proof of work of an Ethereum Virtual Machine (EVM) is presented by Wood [5]; it is the runtime environment for programmable instructions in Ethereum. It provides design and protocol specification using which Ethereum can be implemented. In the recent years, a number of applications have implemented blockchain and Ethereum: IBM uses blockchain for global supply chain management by keeping a record of event data and document workflow for shipping consignments and Ølnes S [6] shows how this technology can be used to enable smart government.

III. DESIGN OF ESCROWCHAIN

A. *Components*

1) Accounts:

These form a major part of an EscrowChain; these indicate the buyers and sellers, ownership details, instructions and the mode of transaction. EscrowChain comprises two types of accounts: Externally Owned Accounts (EOA) and the Contract Accounts that govern the transaction between the users.

Any concerned party such as the buyers, sellers, previous title owners and the community users have Externally Owned Accounts that allows them to participate in a real estate deal. Each of these accounts can only contains a currency or ether, they can send or receive currency, perform actions that can trigger a contract code and also invoke other contracts. Each of the accounts must maintain a minimum ether that can be deducted in the form of gas as mining fees. Any individual or group interested in purchasing a property, residential or commercial is termed as the buyer. Similarly, any individual or group claiming ownership of a property is termed as the seller.

Contract Accounts on the other hand, do not belong to any user but contains code used to govern the transaction in the network; the program logic that controls the real estate deal from beginning to completion. These are lifeless and are active only when a transaction or a message is received putting into motion the execution of the underlying code. These accounts can be said be divided into two types:

- Primary Contracts: Being globally accessible and indestructible, these accounts are used for basic transactions and can be used to create smart contracts specific to a real estate.
- Secondary Contracts: These are a set of specialized contracts that contain logic for the sale of a specific real estate sale and can be accessed only by relevant parties.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

2) Transactions:

It is a signed data package that stores a message to be sent from an externally owned account to any other account on the EscrowChain. It contains details such as the data that determines and restricts the computation cost, the destination, the value being transferred and a security token that acts as the signature of the transaction.

3) States:

The mapping between the account states and the addresses are stored as a trie [16] on every node that participates in the network. The root node of the trie can be used to identify the system state as the structures helps relate the data in the trie to the root which is stored on the blockchain.

B. *Architecture Overview*

EscrowChain is an Ethereum-based distributed ledger protocol that provides a virtual Escrow model for real estate transactions. It contains real estate smart contracts that lie atop the Ethereum blockchain, enabling the creation of a practical Escrow system. The blockchain is used as a communication channel to announce the changes or developments in a real estate deal by appending updates in new blocks, synonymous of an escrow agent tracking the progress of a deal. In EscrowChain, the logic for the implementation of a real estate deal is present in the form of smart contracts, which is a layer on top of Ethereum.

The high level architecture can be divided into three layers as shown in Fig 1. The topmost layer contains the Distributed Applications (DAPP) [8] which allow the buyers and sellers to interact with the system. This is the client facing layer that connects the underlying implementation to the human users outside. The middle layer contains implementation in the form of the Primary and the Secondary Contracts [9]. The users trigger the Primary Contracts that contains the generic logic which in turn call the Secondary Contracts that contains the deal specific logic. The bottom layer is the Ethereum Virtual Machine (EVM) [12] with no modification.

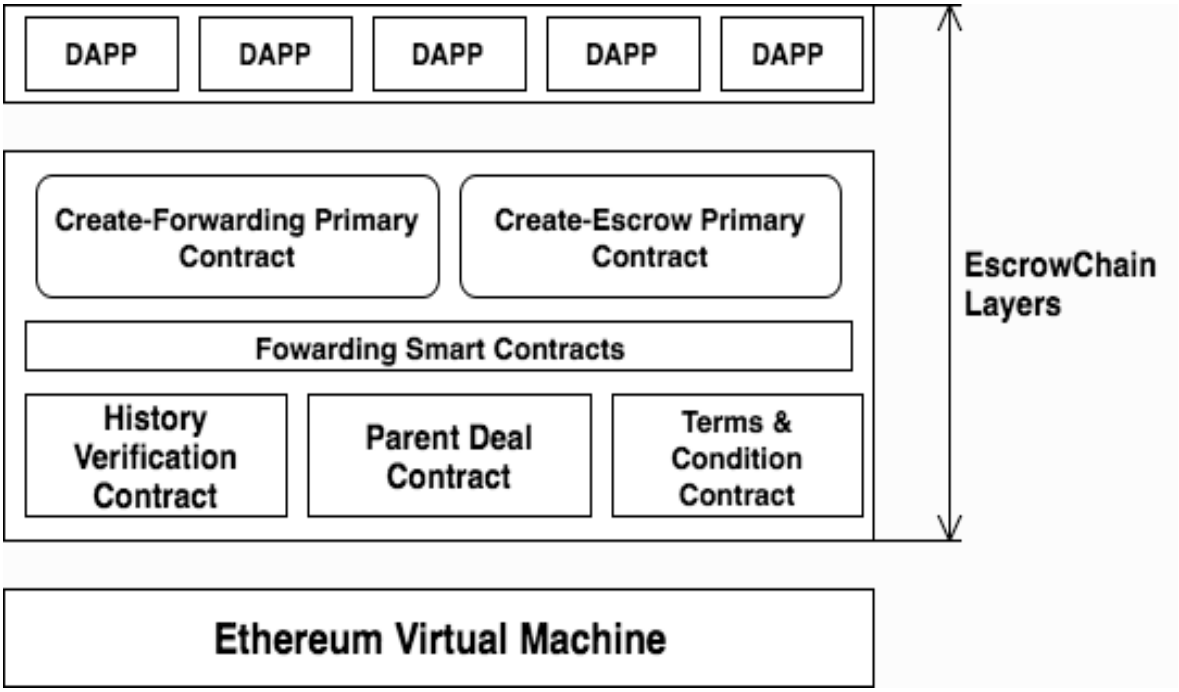

Fig. 1 Architecture of EscrowChain

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

C. *Primary Smart Contracts*

The buyer or the seller can activate a real estate deal using EscrowChain by invoking a Primary Smart contract with relevant parameters. These contracts are globally accessible, are owned by the network and have an immortal lifespan. These contracts in turn generate Secondary Smart contracts specific to the deal. There are two Primary Contracts namely, Create-Forwarding and Create-Escrow.

- Create-Forwarding Primary is invoked by the buyer or the seller and creates the forwarding contract for the users.
- Create-Escrow Primary Contract creates three Escrow Smart Contracts (ESCs) tailored to the real estate deal: Parent Deal Contract, History Verification Contract and Terms and Conditions Contract.

D. *Secondary Smart Contracts*

The contracts that are spawned by the Primary Smart Contracts, contain the logic that is specific to a deal. They can call other Secondary Smart Contracts and Library Smart Contracts. We define four Secondary Smart Contracts to execute a real estate deal namely, Forwarding Smart Contract, Parent Deal Smart Contract, Terms and Conditions Smart Contract and History Verification Smart Contract.

1) Forwarding Smart Contract

The purpose of this contract is to protect the real estate deal from malpractice by decoupling the buyer/seller and the Escrow Smart Contracts (ESCs) by preventing any direct interaction between them. The message from the buyer or the seller is sent to this contract, which is then verified and signed by six users from the community with their respective private keys. This message is then sent to the ESC and the community users are paid in gas [11] for participating in the verification process.

```
contract forwarding_contract {
    /* Define variables and their types  */
    mapping(address => string) public owner;
    string accountType
    uint minSignsRequired
    uint signCount
    string dealID

    function initialization() {
        Set Owner of the contract
        Initialize signCount = 0
        Set whether the this Forwarding Account is for Buyer or Seller
        Call FUNCTION check_Status() and set to dealState
        IF dealState = true
            Generate dealID for the Real Estate Deal
            Send a message to parent_deal_contract
        ELSE
            wait
        ENDIF
    }

    FUNCTION sign() {
        Increment signCount when a user calls this function is called
    }

    FUNCTION check_Status() {
        IF signCount is = 6
           RETURN dealState = true
        ENDIF
     }
}
```

Fig 2. Pseudo code for Forwarding Contract

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

2) Parent Deal Smart Contract

This ESC has two specific tasks: to hold money and documents in escrow and release them in the end to conclude the deal. On behalf of the seller, it holds the title ownership deeds and on behalf of the buyer, the payment amount for the property. By sending messages to the other ESCs, it determines the status of the negotiation and verification. When the status indicates that all the obligations by the buyers and sellers are met, it transfers the money and ownership to the respective party.

```
contract parent_deal_contract {
    /* Define variables and their types */
    uint dealState
    private string dealID
    string sellerDocuments
    uint saleAmount

    FUNCTION initialization(string msg){
        Set dealID = dealID in msg
        Set sellerDocuments sent by Seller
        Set saleAmount sent by Buyer
        Call history_verification_contract
        Call terms_condition_contract
        Call FUNCTION check_Deal_State() and set to dealState
        IF dealState = true
            Call Library FUNCTION transferDocumentToBuyer()
            Call Library FUNCTION transferAmountToSeller()
        ELSE
            Call Library FUNCTION transferDocumentToSeller()
            Call Library FUNCTION transferAmountToBuyer()
        ENDIF
    }

    FUNCTION check_Deal_State(){
        Get the historyStatus from history_verification_contract
        Get the dealStatus from terms_condition_contract
        IF historyStatus = true AND dealStatus = true
            RETURN dealState = true
        ELSE
            RETURN dealState = false
        ENDIF
    }
}
```

Fig 3. Pseudo code for Parent Deal Smart Contract

3) History Verification Smart Contract

The blockchain ledger contains historical data the property on sale. Six community users participate in the verification of the title details and sign the forwarding contract with their private keys respectively. The message containing status of this validation is sent to the Parent Deal Smart Contract.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

```
contract history_verification_contract {
    /* Define variables and their types */
    bool verificationState = false
    private string dealID
    string[] memory prevOwners = new string[](25)
    string[] memory repairs = new string[](25)
    string[] memory crimeRecord = new string[](25)
    string[] memory disasters = new string[](25)
    uint signCount

    FUNCTION initialization(string msg){
        Set dealID = dealID in msg
        FOR i = 1 to 25 step 1 DO
            Set prevOwners[i] = prevOwners in msg
        ENDFOR
        FOR i = 1 to 1000 step 1 DO
            Set repairs[i] = repairs in msg
        ENDFOR
        FOR i = 1 to 10 step 1 DO
            Set crimeRecord[i] = crimeRecord in msg
        ENDFOR
        FOR i = 1 to 10 step 1 DO
            Set disasters[i] = disasters in msg
        ENDFOR
        Get the value of signCount
        IF signCount > 6
            Set verificationState = true
        ELSE
            Set verificationState = false
        ENDIF
    }

    FUNCTION sign() {
        Increment signCount when a user calls this function is called
    }
}
```

Fig 4. Pseudo code for History Verification Smart Contract

4) Terms and Conditions Contract

The terms and conditions set forward by the two parties involved are maintained by this ESC. This smart contract allows the buyer and seller to enter into a negotiation about the deal. Finally, after suggesting edits to the conditions both the parties may choose to accept or reject the deal. The status of acceptance or rejection of the agreement is then sent to the Parent Smart Contract.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

```
contract terms_condition_contract {
    /* Define variables and their types */
    bool buyerNegotiationState = false
    bool sellerNegotiationState = false
    bool dealState = false
    private string dealID

    FUNCTION initialization(string msg){
        Set dealID = dealID in msg
        IF sellerFlag != 1
            sellerNegotiationState = false
            Send msg to Seller that sellerTerms has been not been accepted and to modify
        ELSE
            sellerNegotiationState = true
        ENDIF
        IF buyerFlag != 1
            buyerNegotiationState = false
            Send msg to Buyer that buyerTerms has been not been accepted and to modify
        ELSE
            buyerNegotiationState = true
        ENDIF
        IF buyerNegotiationState = true AND sellerNegotiationState = true
            Set dealState = true
            Send msg to parent_deal_contract
        ELSE
            wait
        ENDIF
    }

    FUNCTION buyer_agreement_state(string msg) {
        Buyer Calls this FUNCTION to agree or disagree to Sellers Term
        FOR i = 1 to noOfTerms step 1 DO
            IF agreement in msg = sellerTerms[i]
                Set seller_Term_agreed[i] = true
                Send msg to Seller that sellerTerms[i] has been accepted
            ELSE
                Set seller_Term_agreed[i] = false
                Set sellerFlag = 1
                Send msg to Seller that sellerTerms[i] has been rejected
            ENDIF
        ENDFOR
    }

    FUNCTION seller_agreement_state(string msg) {
        Seller Calls this FUNCTION to agree or disagree to Buyers Term
        FOR i = 1 to noOfTerms step 1 DO
            IF agreement in msg = buyerTerms[i]
                Set buyer_Term_agreed[i] = true
                Send msg to Buyer that buyerTerms[i] has been accepted
            ELSE
                Set buyer_Term_agreed[i] = false
                Set buyerFlag = 1
                Send msg to Buyer that buyerTerms[i] has been accepted
            ENDIF
        ENDFOR

    }
}
```

Fig 5. Pseudo code for Terms and Conditions Smart Contract

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

IV. PROPOSED METHODOLOGY

A real estate purchase using EscrowChain can be achieved in three distinct phases: deal creation, execution and finalisation. The complete process can be expedited as all the parties transact online and the verification is handled by the community users on the network.

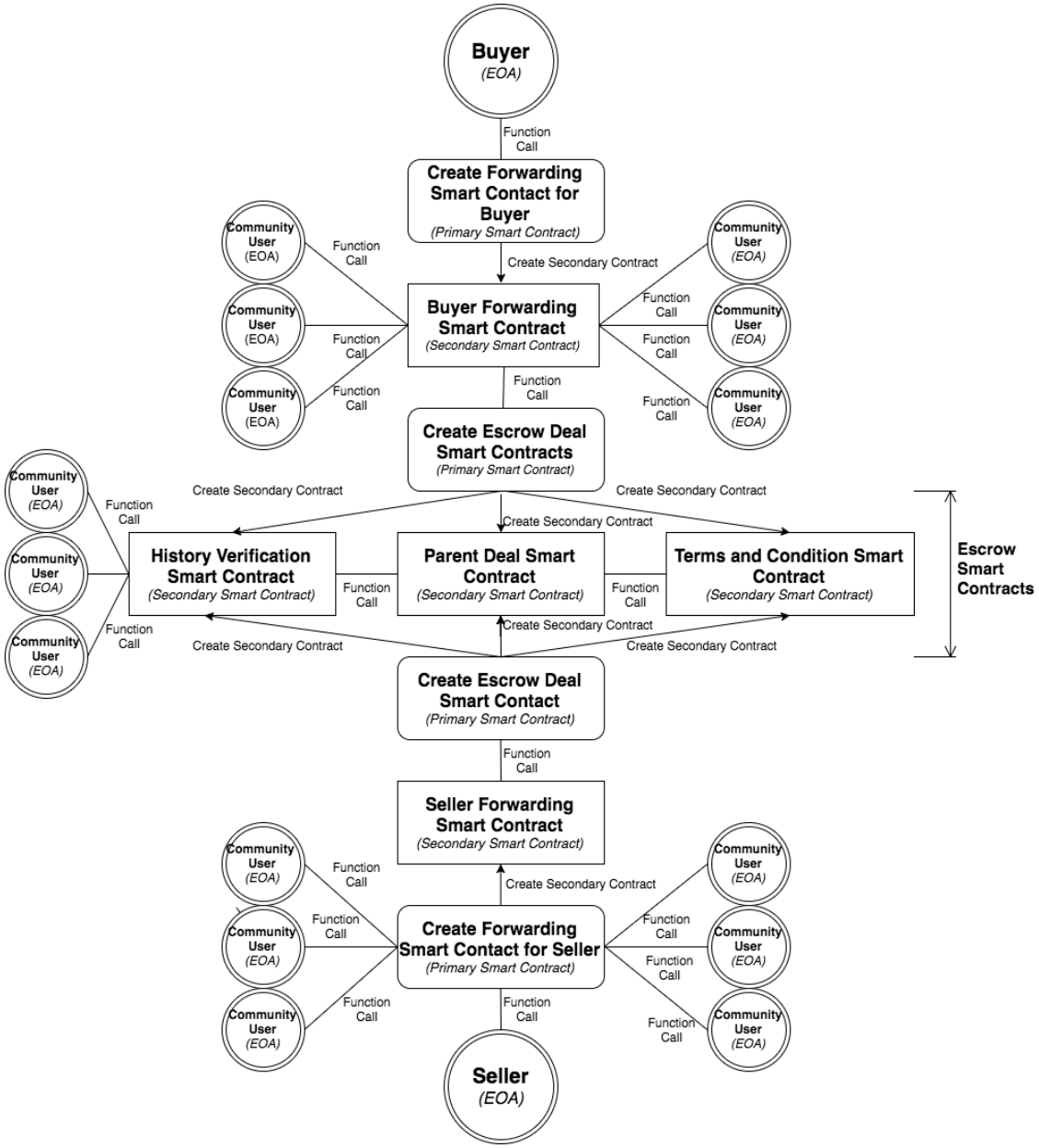

Fig 6. Implementation of EscrowChain

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

A. *Deal Creation Phase*

The deal can be initiated by either the buyer or the seller. Both parties, in no particular order create or use an existing account on the blockchain network, changing the state if a new EOA is created. The buyer then triggers the Create-Forwarding primary smart contract, which in turn creates the Forwarding smart contract for the buyer. The seller likewise, gets a Create-Forwarding contract followed by the Forwarding smart contract. The state change indicates the creation of contract accounts.

After the creation of both types of accounts by the buyer as well as the seller, either party can trigger the creation of the three Escrow Smart Contracts (ESCs) using a digitally signed token that acts as the secret code for the transaction. After the state change reflects this, the party that created the ESC shares the secret code with the other party. Upon receiving the secret code the user triggers the creation of ESC, but instead of new contracts gets connected to the existing ones. The state at the end of this phase indicates that the both parties have entered the deal.

B. *Deal Execution Phase*

The buyer can now transfer the sale amount via the Forwarding Smart Contract, which will be held in Escrow. The community users start the verification; on approval a message is sent to the Parent Deal Smart Contract, on failure the money is returned to the buyer after deducting the Gas equal to the fees incurred during mining. The state is updated to reflect this change. When the buyer initiates the transfer of money, the seller can also submit the ownership documents via the Forwarding Smart Contract. If the property is offered for sale for the first time using EscrowChain, the property gets associated to a unique hash called the propertyId, otherwise the owner need only submit the propertyId. The community users again participate in the process of validating the authenticity of the documents and verify if it belongs to the owner as claimed. When the community approves the document to be genuine, the Parent Deal smart contract is updated using a message. Mining fees is deducted from the owner's account and the Parent Deal is updated with the status of the payment and document verification.

The parties then enter into negotiations and the agreement is placed in the Terms and Conditions contract via the respective Forwarding contracts. The buyers and sellers can agree, disagree or suggest edits to the agreement. In the meantime, the the History Verification smart contract allows community users to start the verification of the metadata using the propertyId; previous owners or tenants, repairs, criminal records and other miscellaneous details are verified. The Terms and Conditions contract and History Verification contract send an independent status to the Parent Deal contract. The state at the end of this phase indicates that the buyer and the seller have started negotiations and each negotiation is recorded in the transaction trie.

C. *Deal Finalisation Phase*

Upon the receipt of positive status from both the contracts, the Parent Deal Contract calls the Library contracts; transferAmountToSeller() and transferDocumentToBuyer(). When the money is released and the ownership to the propertyId updated, the deal comes to a closure. If a negative status is received from either the History Verification contract or the Terms and Condition contract, Parent Deal Contract calls the Library contracts; transferAmountToBuyer() and transferDocumentToSeller() returns the money and documents to the respective owners and the deal is cancelled. At the end of this phase, the series of transactions are recorded in a new block indicating the updated owner of title and money for a successful deal or cancelled deal. This block is finally added to the blockchain.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

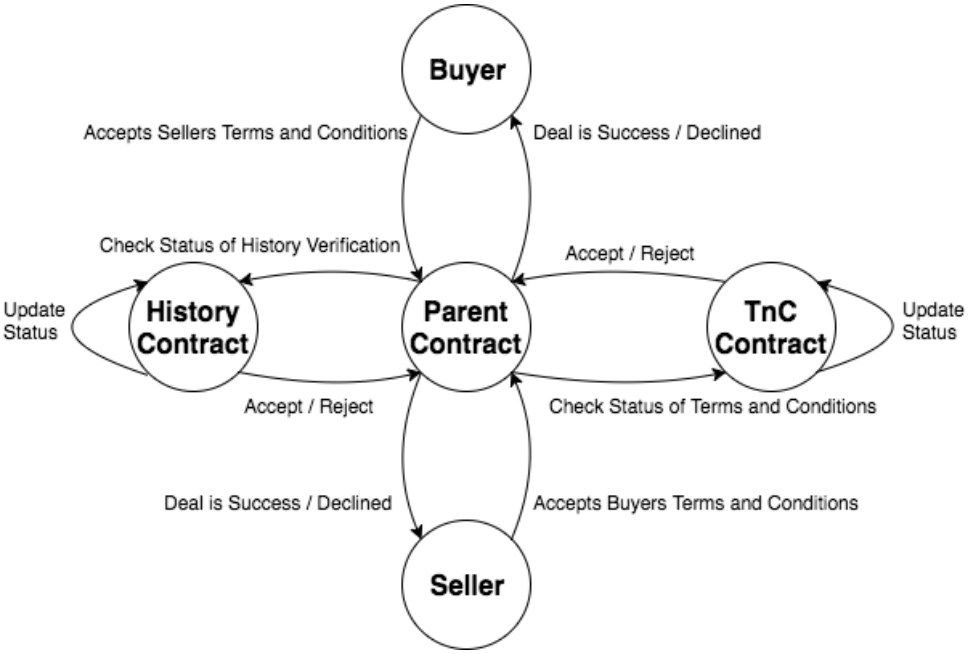

Fig 7. State Transition Diagram of EscrowChain

V. SECURITY

The smart contract and its state data along with transaction data should be only exposed to the concerned parties in the network; exposed to the outside world, the real estate deals can become vulnerable to attacks and insecure property transactions. EscrowChain inherits the security implementation of Ethereum [13][17] and add to it.

Any programming logic can be added as a smart contract to Ethereum and the program can run infinitely, taking up a lot of computational power and resources. Ethereum prevents this by setting up a threshold on the amount of gas that can be used for the computation; every transaction has a gasLimit and gasPrice which determines number of computational steps a contract will take upon execution and on reaching the gasLimit, the contract execution stops.

Ethereum ensures that a contract execution runs until completion and cannot stop midway. In any event where the available Gas for a Transaction is exhausted or the contract execution gets interrupted, it reverts the transaction to the previous state, thus preventing the existence of any intermediate state which attackers can exploit.

To prevent any Replay Attacks, Ethereum uses a unique incrementing value “nonce” for each transaction which does not allow any party to reinsert blocks to the ledger.

Each transaction is also followed by a Transaction Receipt which is saved in the block as a hash. The receipt contains a hash representing the post-transaction state and cumulative gas used in block containing the transaction receipt. This enables step by step audit and validation of individual blocks in the blockchain which can help identifying malicious transactions.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 10, October 2017

VI. CONCLUSION

We have proposed a design of EscrowChain, using which any real estate deal can be created, negotiated and closed digitally without the requirement of an escrow organisation and paperwork. The users signed up on the network can rely on the existing blockchain technology to record the transaction, the shared network for faster verification and smart contracts to handle the process. We have implemented a set of smart contracts to hold the money and documents in escrow allowing the users to negotiate and transfer the ownership and payment. The design is transparent and the user is assured of fraud free transaction.

VII. ACKNOWLEDGEMENT

We are immensely grateful to Dr.Asha T, Professor, Department of Computer Science & Engineering, Bangalore Institute of Technology for the help, guidance and valuable feedback given during the entire course of our research work.

REFERENCES

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1:2012, 2008.
[2] Vitalik Buterin. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. 2013a. URL. Available: http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
[3] Meni Rosenfeld. Overview of Colored Coins. 2012. URL Available: https://bitcoil.co.il/BitcoinX.pdf
[4] Nick Szabo. Formalizing and securing relationships on public networks. First Monday, 2(9), 1997
[5] Wood, Gavin. "Ethereum: A secure decentralized generalised transaction ledger." *Ethereum Project Yellow Paper* 151 (2014).
[6] Ølnes, Svein. "Beyond bitcoin enabling smart government using blockchain technology." *International Conference on Electronic Government and the Information Systems Perspective*. Springer International Publishing, 2016.
[7] Ali, Muneeb, et al. "Blockstack: Design and implementation of a global naming system with blockchains." *Last visited on* 25.2 (2016).
[8] Bogner, Andreas, Mathieu Chanson, and Arne Meeuw. "A decentralised sharing app running a smart contract on the ethereum blockchain." Proceedings of the 6th International Conference on the Internet of Things. ACM, 2016.
[9] Kosba, Ahmed, et al. "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts." Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016.
[10] Wang, Luqin, and Yong Liu. "Exploring miner evolution in bitcoin network." *International Conference on Passive and Active Network Measurement*. Springer, Cham, 2015.A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[11] Dannen C. (2017) Mining Ether. In: Introducing Ethereum and Solidity. Apress, Berkeley, CA
[12] Hildenbrandt, Everett, et al. *KEVM: A Complete Semantics of the Ethereum Virtual Machine*. 2017.
[13] Gervais, Arthur, et al. "On the security and performance of proof of work blockchains." *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016.
[14] Olson, A. "Escrow management structure." U.S. Patent Application No. 10/777,344.
[15] Swanson, Tim. "Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems." *Report, available online, Apr* (2015).
[16] Willard, Dan E. "New trie data structures which support very fast search operations." *Journal of Computer and System Sciences*28.3 (1984): 379-394
[17] Li, Xiaoqi, et al. "A survey on the security of blockchain systems." *Future Generation Computer Systems* (2017).