Efficient and Secured Transactions of Data in Crucial Network using Blockchain Technology

Journal:	*Transactions on Cyber-Physical Systems*
Manuscript ID	Draft
Manuscript Type:	Paper for User-Centric Security and Safety for CPS
Date Submitted by the Author:	n/a
Complete List of Authors:	Athreya A, Manoj; Vidyavardhaka College of Engineering, Computer Science & Engineering Kumar, Ashwin; Vidyavardhaka College of Engineering, Computer Science & Engineering S M, Nagarajath; Vidyavardhaka College of Engineering, Computer Science & Engineering H L, Gururaj; Vidyavardhaka College of Engineering, Computer Science & Engineering
Keywords:	Blockchain, Decentralized, Distributed, Smart contracts, peer-to-peer network

Efficient and Secured Transactions of Data in Crucial Network using Blockchain Technology

Manoj Athreya A
Computer science & Engineering

Vidyavardhaka College of Engineering
Mysuru, India
manojathreya13@gmail.com

Ashwin A Kumar
Computer science & Engineering

Vidyavardhaka College of Engineering
Mysuru, India
ashwinkumar3098@gmail.com

Nagarajath S M
Computer science & Engineering

Vidyavardhaka College of Engineering
Mysuru, India
nagarajath98@gmail.com

Gururaj H L
Assistant Professor,Computer science & Engineering

Vidyavardhaka College of Engineering
Mysuru, India
gururaj1711@vvce.ac.in

***Abstract*— Blockchain is a growing list of records called blocks, that are linked using cryptography. It is a decentralized, distributed and an immutable ledger to store digital transactions. Its databases are managed using peer-to-peer network where all the nodes in a network are equal and is the major concern in the types of network architecture. The decentralization of blockchain means that it won't depend on a central point of control. With a lack of single authority, which makes the system equitable and more secure to validate transactions and to record data which makes it incorruptible. It uses consensus protocol which is a set of rules that describes the communication and transmission of data between nodes, to transact securely with other users without relying on the central authority. In this paper, we have proposed an innovative and efficient way of adopting pets using Blockchain technology, the adoption of pets and its payment method is made more easier and assures high security to data. Many approaches are made to the application but the approach using decentralized system is not used and it provides a new dimension to the application.**

Keywords— Blockchain, Decentralized, Distributed, peer-to-peer network, Smart contracts.

I. Introduction

Blockchain is a set of linked-lists where the data stored in it is immutable, which means once the data is stored it cannot be changed. It is decentralized, distributed and public digital ledger. Decentralization means storing data in different nodes across peer-to-peer network, thus eliminating the risks when the data is stored centrally. A blockchain, also called as a distributed, immutable ledger is essentially an append-only data structure maintained by set of nodes which won't trust each-other fully. Nodes in a blockchain network agree on a set of blocks which are ordered, having multiple transactions. Hence, blockchain is viewed as a log of ordered transactions [8]. Blockchain enabled smart contracts employ proof-of-stake validation for transactions, which promises significant performance advantages compared to proof-of-work solutions [12]. Blockchain technology aims at increasing magnitude of flexible traffic of evolving complexity. The target is to allow complex services which are secure, sustainable and efficient. Therefore, the target is to accelerate/scale blockchain functionality [13]. Smart contracts have become a reality with the boom of blockchain technology, which operates without trusted third parties for settling transactions and disagreements among pseudonymous participates [1]. The blockchain enforced smart contracts maintain lawful and monetary services, which are currently ineffective and likely unsustainable [13]. Peer-to-Peer (P2P) protocols provides distribution of high data capacity to users, which are scalable [10]. Peer-to-peer blockchain networks has no central point of failure, as they lack centralize points of vulnerability that hackers can exploit. Blockchain technology can be used to create a constant, transparent, public ledger for organizing sales records, which tracks digital usage and payments distribution to content creators. [6]. Ethereum is an essential and an ultimate foundation layer, where a blockchain with in-built Turing-complete programming language. It allows anybody to write smart contracts for building decentralized applications with their own rules of state functions, transactions, and ownership [1]. A distributed ledger is a set of linked-lists that consists of an ordered list of transactions. Where a ledger may contain transactions of money done through banks or property exchanged between known parties. A distributed ledger in a blockchain is duplicated over the nodes in a network. Solidity is a high-level, contract-oriented language which helps in writing smart contracts. It mainly influences Python, C++ and JavaScript and was built for Ethereum Virtual Machine (EVM) [1].

1)**Getting Trust:** In Ethereum network where the system runs in a fully decentralized manner, consensus mechanism is used to solve problems about guaranteeing truthful service. The facticity of services provided to the network is automatically checked and guaranteed by the consensus. Providers who cheat will be punished or even kicked out from the system. Success of decentralization depends not only on peer-to-peer network but also on being trustless, where an environment needs no trust and no centralization.

2)**Protecting Security**: Blockchain system and a peer-to-peer network form the secure backbone of the Ethereum network. Communication operations between devices are embedded in the blockchain and service information is stored and routed by peer nodes and guaranteed by the consensus. The system therefore has no central point which is exposed to attackers and security is guaranteed by the consensus mechanism of distributed miners who are economically motivated to be honest.

3)**Achieving Fairness:** In Ethereum, services are published/subscribed to/from the network. The matching process is performed by a smart contract which cannot be controlled any end-point or a centralized party. The smart contract runs a Truthful Continuous Double Auction (TCDA) that prevents cheating and maximizes social welfare of the entire community. TCDA is mathematically strategy-proof, where there is no incentive for traders to lie or hide their personal information from other traders. This is blockchain based technology to approach to the global fairness of service distribution in system.

4)**Strengthening Incentive:** The real-time analysis is carried out for identifying supply and demand, where Ethereum creates new marketplaces built on the foundation established by system and blockchain network for transformation. This enables new peer-to-peer model of economies. In the next section, we will explain briefly about blockchain and its features. Section 3.0 explains the algorithm incorporated. Section 4.0 deals with the details of the project creation and its working. Section 5.0 concludes.

II. Related Work

In earlier days, people used to maintain a ledger to store data systematically. When it comes to maintaining financial transactions, it is more important and must be secured. After digitalization the paper format was eliminated and data was stored in computers by using programs. These ledgers were maintained by a central authority which is bank or government which act as a trusted party. These digital ledgers were stored and maintained in a central authority through servers. Hence these centralized systems became an attractive point for adversaries' party they experienced single point of failure. These stored data are hackable with the conventional security being provided. Therefore, Blockchain technology came into picture to prevent security vulnerabilities. Blockchain technology is based on an idea of immutable ledger to maintain all the records within their network. It prevents single point of failure and blockchain distributes the ledgers to every node in the network, so that every node can see what is happening, and if suspicious activity occurs at one node, other nodes in the network will be notified

A. History

Wei Dai, in 1998 became the first person to present the paper which introduced the idea of incorporating virtual money through working out cryptographic puzzles as well as decentralized consensus, but his attempt failed as he did not provide adequate details on how decentralized consensus worked. Hal Finney, in 2005 presented a concept of "Proof of Work", a network which uses idea of solving Hash cash puzzles to create a concept of cryptocurrency, but did not concentrate on issue for trusted computation on backend. Satoshi Nakamoto, in January 2009 set the first blockchain into motion in the name of bitcoin which concurrently introduced two radical and untested concepts, a decentralized peer-to-peer online currency and the second is the idea of a proof of work blockchain to allow for public agreement on the order of transactions. The invention of this idea by Satoshi is simple decentralized consensus protocol, which is based on peers combining transactions into block in regular intervals forming a ever growing blockchain. With proof of work mechanism, peers have the right to participate in the system.

B. Blockchain

Nowadays, Bitcoin's technology is seeking more attention towards its second part and how blockchain concept can be used for just more than money and transaction. There are other applications that has incorporated blockchain. Name coin is a service which represent decentralized name registration database which provides way of identifying accounts so that other people can interact with them and allowing users to create their own cryptocurrency. Another service called Colored coins which is a more advanced application providing decentralized exchange, financial derivatives, peer-to-peer gambling and on-blockchain identity. Thus, building an independent network and building a protocol on top of Bitcoin are the current trends on blockchain.

C. Block

A block in a blockchain consists of two main parts block's header and body with fields block version, nonce, data, timestamp, previous hash value and the hash value of the current block. It is a data structure with ordered timestamp. The data consists of transactions details. The transactions to be stored in each block is determined by the block size and the size of transaction.

D. Decentralization

Decentralization is one of the main and unique features of blockchain technology where it distributes the data to every node present in the network. As it is decentralized, it prevents single point of failure. Besides, it is an immutable data where hackers have to change every node in the blockchain network.

E. Digital Signature

In a blockchain network every user owns a pair of public and private key. The digital signature involves two phases: signing and verification phase, where the private key is used to sign digital transactions and hence it is to be kept confidential. Public key is used for transaction between two parties.

F. Consensus Algorithm

Every node in a blockchain network needs to ensure that the data present in its node and the data distributed over the nodes in a network are identical. Thus, it uses the consensus algorithm to agree on one constant state of blockchain transaction. There are several approaches for this algorithm such as Proof-of-Work (PoW), Proof-of-Stake (PoS), Byzantine Fault Tolerance, Delegated Proof-of-Stake (DPoS), Tendermint and Ripple.

G. Limitations of Scripting language implemented in Bitcoin

➢ Lack of Turing-completeness - Bitcoin does not support loops in transaction verification.

➢ Value-blindness - UTXO script does not provide the option of withdraw capability (unspent transaction outputs or UTXO)

➢ Lack of state and Blockchain-blindness - UTXO does not allow decentralized exchange and are blind to blockchain data parameters such as nonce and hash value.

III. PROOF OF WORK ALGORITHM

In the blockchain, there is a high degree of transparency which can be used in many areas. To achieve transparency in the system several algorithms are proposed such as proof of work(PoW) and proof of stake(PoS).

Proof-of-Work is the major consensus algorithm in a Blockchain network. This algorithm is mainly used to confirm and validate transactions and mine new blocks to the existing chain. With PoW, miners compete to find the hash value of the block and facilitate the completion of transactions on the network and get rewarded. The target value or the nearest nonce value of the block to be mined by the miners is recalculated and updated every 2016 blocks.

The Mining Algorithm is as follows:

- Receive the hash of the previous block from the network.
- Retrieve a list of transactions which is known as a block. The list comes from the bitcoin network.
- Determine the hash value of a block with a random number.
- If the hash value is more than the current difficulty level, then the block will be mined. If not, start over from Step 1.

The pseudocode for the above PoW algorithm:

P = Hash of the previously mined block

B = Block which consists of transactions

H = Hash function

D = Difficulty Level

```
1. Receive P
2. Construct B
3. Apply H(P, B, Random Number) . IF RESULT > D END
4. GOTO 2
```

By this mining algorithm, miners use special softwares to find the nonce or hash value to mine a block containing transactions.This provides a smart way to issue the currency and keep the network secure Abbreviations and Acronyms.

IV. CASE STUDY ANALYSIS

The most remarkable thing about this Blockchain technology is that it increases the capacity of the whole network and keeps its ledgers in a never-ending state of forwarding momentum making it immutable. It is built in such a way that any transaction added to the block or a block added to the chain cannot be altered which provides high contrast of secureness. It is very easy to pinpoint any race problem and debug if there is any. Format of it is designed so that it can also create a unchangeable trail. This technology is extremely secure as every individual who gets in the blockchain network is given with unique identification address which is connected to his account in Metamask.This guarantees that the account holder himself is handling the transactions. The encryption of block in chain makes it hard-boiled for any hacker to strike the existing setup of the chain. Transaction speed is increased to very high level as existing banking organisation takes plenty of time to process and initiate transaction.

Ethereum is preferred as a better platform for development and blockchain network. Extending the network which provides wide extent of use cases, which is powered by smart contracts. All transactions are done in real time and for some exchange of Ethers (currency of Ethereum network) , all blocks area scripted by miners, who execute these scripting and validation transaction, which is expensive with respect to energy and time.

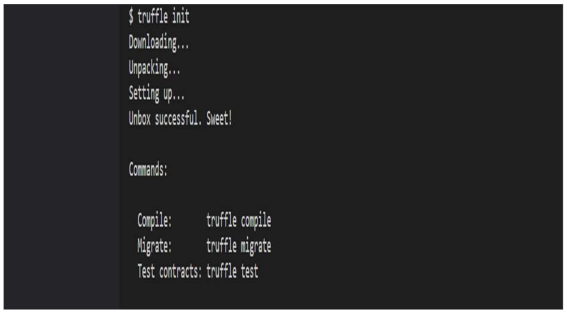
Smart Contracts help user to adopt pet from one buyer by paying them through Metamask and ensure the transaction. Language which is used to write smart contract is Solidity, which is a combination of both JavaScript and C++. They are executed by the peers in Ethereum network for every 15 seconds and validate again with at least 2 other peers to activate it. After that, methods of contracts are executed and are shared among other peers in the network. After they are initialised, they would not be able to discard from blockchain, and peers can look at their result whether the execution of smart contract is true or not. Web3js is a lightweight JavaScript library which is used to integrate with clients (nodes) on the decentralized network. It basically a communicator between blockchain and smart contracts. Testing and Validation is done using Solidity.

Use of real Ethereum network is quite expensive for experimenting, developing and testing purpose (requires spending of Ethers) . Also, it occupies huge memory in network. Thus, private Test networks are used and made available to the developers and one of them is Ganache, a personal blockchain for Ethereum development you can use to deploy contracts, develop your applications, and run tests. In order to extend your test network, peers should download a legit wallet from the official website and change the connection with main network to preferred test network, using Metamask.

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

A. *Creating A Truffle Project*

Create a bare Truffle project using truffle init (fig 4.1.1). This command unboxes the project with no smart contracts.



```
$ truffle init
Downloading...
Unpacking...
Setting up...
Unbox successful. Sweet!

Commands:

  Compile:        truffle compile
  Migrate:        truffle migrate
  Test contracts: truffle test
```

fig4.1.1

B. *Folder Structure*

The default Truffle folder structure contains:

- contracts/: This folder contains the source files for our smart contracts.
- migrations/: Truffle has a special feature that keeps track of smart contract changes made and uses a migration system to handle smart contract deployments.
- test/: Test for smart contract are written both JavaScript and Solidity
- truffle.js: Configuration file for truffle

C. *Writing a Smart Contract*

C.1 Migration.sol

In order for truffle to migrate smart contract to the network, Truffle requires to have a migration contract. This contract contains a specific interface for developers to edit and change according to necessities. Contract will be initialized and won't be edited again. This .sol file is created during unboxing of project. (fig 4.3.1.1)

C.2 Adoption.sol

Solidity is a statically-typed language and has a unique type called an address. which is 20byte values where every account and smart contract on Ethereum network are stored. Ethereum blockchain uses address type to receive and send Ethereum tokens to and from the peers. Adoption contract allow users to make adoption requests and retrieving the adopters. (fig 4.3.2.1)

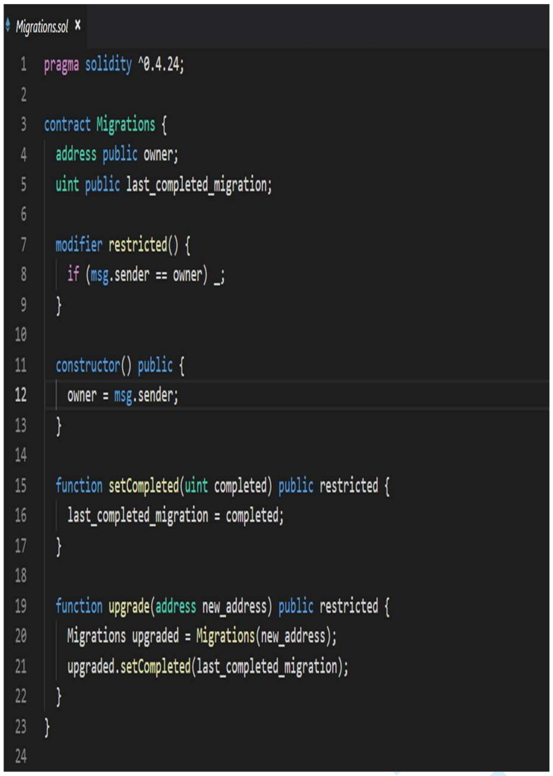
D. *Compiling and migrating the Smart Contract*

Solidity is a compiled language which is similar to translating human readable language to solidity into something that Ethereum Virtual Machine (EVM) understands. (fig 4.4.1)

A migration is a deployment script used to change the state of your application's smart contracts, moving it from one phase to the next. First time, migration are just being deployed as new code, but later, other migrations might replace the contract with a new one.

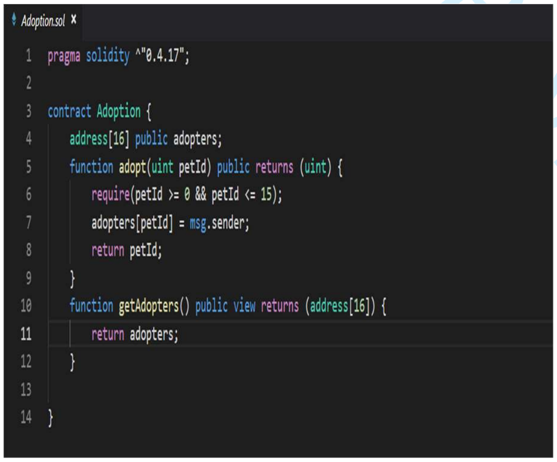
1_initial_migration.js and 2_deploy_contracts.js deals with deploying Migrations.sol contract to sense smart contract migrations and ensures that they don't double migrate unchanged contracts (reset flag allow us to add/upgrade contracts and redeploy the affected contracts only) (fig 4.4.2)

In Ganache, state of the blockchain has changed and blockchain now shows that the current block, where previously it was 0, now it is 4.(fig 4.4.3)



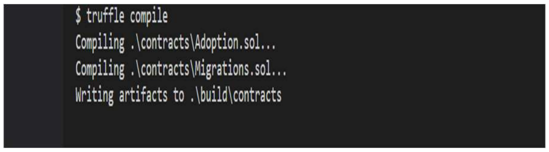
```
pragma solidity ^0.4.24;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  constructor() public {
    owner = msg.sender;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }

  function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}
```

fig 4.3.1.1



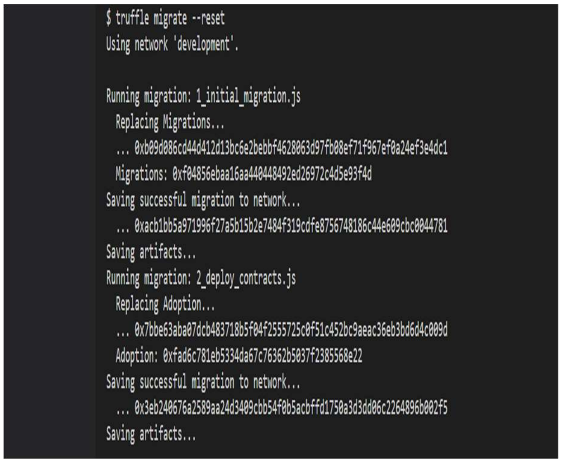
```
pragma solidity ^"0.4.17";

contract Adoption {
    address[16] public adopters;
    function adopt(uint petId) public returns (uint) {
        require(petId >= 0 && petId <= 15);
        adopters[petId] = msg.sender;
        return petId;
    }
    function getAdopters() public view returns (address[16]) {
        return adopters;
    }

}
```

fig 4.3.2.1



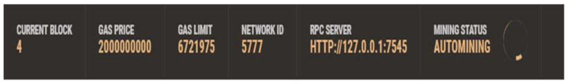
```
$ truffle compile
Compiling .\contracts\Adoption.sol...
Compiling .\contracts\Migrations.sol...
Writing artifacts to .\build\contracts
```

Fig 4.4.1



```
$ truffle migrate --reset
Using network 'development'.


Running migration: 1_initial_migration.js
  Replacing Migrations...
  ... 0xb09d086cd44d412d13bc6e2bebbf4628063d97fb08ef71f967ef0a24ef3e4dc1
  Migrations: 0xf04856ebaa16aa440448492ed26972c4d5e93f4d
Saving successful migration to network...
  ... 0xacb1bb5a971996f27a5b15b2e7484f319cdfe8756748186c44e609cbc0044781
Saving artifacts...
Running migration: 2_deploy_contracts.js
  Replacing Adoption...
  ... 0x7bbe63aba07dcb483718b5f04f2555725c0f51c452bc9aeac36eb3bd6d4c009d
  Adoption: 0xfad6c781eb5334da67c76362b5037f2385568e22
Saving successful migration to network...
  ... 0x3eb240676a2589aa24d3409cbb54f0b5acbffd1750a3d3dd06c2264896b002f5
Saving artifacts...
```

Fig 4.4.2



CURRENT BLOCK 4

GAS PRICE 2000000000

GAS LIMIT 6721975

NETWORK ID 5777

RPC SERVER HTTP://127.0.0.1:7545

MINING STATUS AUTOMINING

Fig 4.4.3

E. UI to interact with Smart Contract

Developers use web3 to interact with Ethereum, where there instantiation of smart contract is done so web3 knows where to find it. Truffle has a library to help which is called truffle-contract. It stores all information about the contract in sync with migrations, manual changes of contract are not required each time.

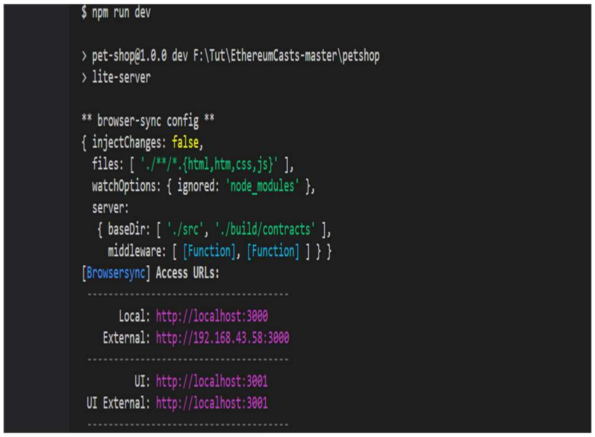
Retrieve the artifact file for the smart contract and Once it is done, callback function pass them to Truffle Contract () so that it creates an instance of the contract which enables to interact with contract that has been instantiated, App.web3Provider which is set as web3 provider where values are stored earlier while setting up web3.

F. Interacting with the Dapp Browser

The easy way to interact with decentralized app in a browser is through Metamask, which is a browser extension available for popular browsers like Chrome and Firefox. Connect Metamask to the blockchain created by Ganache by editing configuration file of truffle.

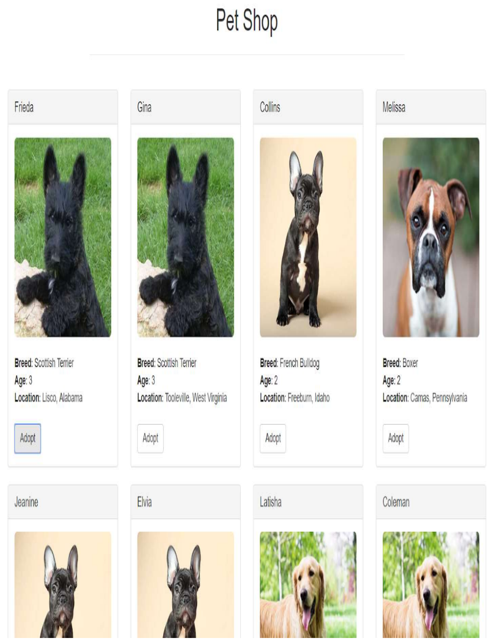
lite-server library is used to serve our static files and development server will launch automatically by opening a new browser tab containing your dapp. (fig 4.6.1)

On Clicking the Adopt button on the pet of choice (fig 4.6.2), a Metamask prompt will automatically open seeking to approve the transaction (fig 4.6.3). Click Submit to approve the transaction. This would add a new transaction in the blockchain returning the transaction hash(fig 4.6.4)



```
$ npm run dev

> pet-shop@1.0.0 dev F:\Tut\EthereumCasts-master\petshop
> lite-server

** browser-sync config **
{ injectChanges: false,
  files: [ './**/*.{html,htm,css,js}' ],
  watchOptions: { ignored: 'node_modules' },
  server:
   { baseDir: [ './src', './build/contracts' ],
     middleware: [ [Function], [Function] ] } }
[Browsersync] Access URLs:
 --------------------------------------
       Local: http://localhost:3000
    External: http://192.168.43.58:3000
 --------------------------------------
          UI: http://localhost:3001
 UI External: http://localhost:3001
 --------------------------------------
```

Fig 4.6.1



Pet Shop

Frieda
Breed: Scottish Terrier
Age: 3
Location: Lisco, Alabama
Adopt

Gina
Breed: Scottish Terrier
Age: 3
Location: Tooleville, West Virginia
Adopt

Collins
Breed: French Bulldog
Age: 2
Location: Freeburn, Idaho
Adopt

Melissa
Breed: Boxer
Age: 2
Location: Camas, Pennsylvania
Adopt

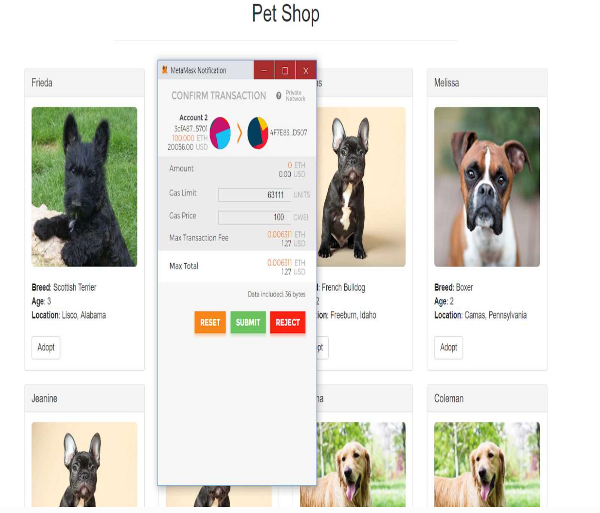
Jeanine

Elvia

Latisha

Coleman

Fig 4.6.2



Pet Shop

MetaMask Notification

CONFIRM TRANSACTION
Private Network

Account 2
3cfA87...5701
100.000 ETH
20056.00 USD
4F7E83...D507

Amount 0 ETH 0.00 USD
Gas Limit 63111 UNITS
Gas Price 100 GWEI
Max Transaction Fee 0.006311 ETH 1.27 USD
Max Total 0.006311 ETH 1.27 USD

Data included: 36 bytes

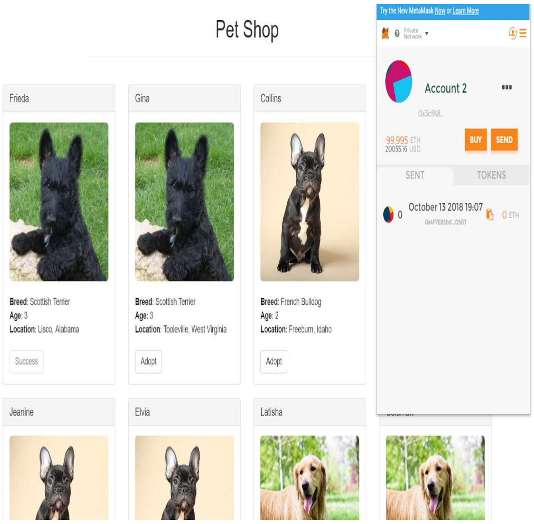
Frieda
Breed: Scottish Terrier
Age: 3
Location: Lisco, Alabama
Adopt

Melissa
Breed: Boxer
Age: 2
Location: Camas, Pennsylvania
Adopt

Jeanine

Coleman

Fig 4.6.3



Pet Shop

Frieda
Breed: Scottish Terrier
Age: 3
Location: Lisco, Alabama
Success

Gina
Breed: Scottish Terrier
Age: 3
Location: Tooleville, West Virginia
Adopt

Collins
Breed: French Bulldog
Age: 2
Location: Freeburn, Idaho
Adopt

Jeanine

Elvia

Latisha

Try the New MetaMask Now or Learn More

Private Network

Account 2
0x3cfA8...
99.995 ETH
20055.16 USD

0 October 13 2018 19:07 0x4F7E83bd...D507 0 ETH

Fig 4.6.4

V. CONCLUSION

In this paper, as proposed we have used a different technology stack for reliable and user-friendly interactions with the system. Our architecture uses blockchain technology which assures more security to the data as it is stored in blocks using cryptography, intern these blocks are stored in nodes of different computers in a network. The focus is mainly on components that strengthen the functionality of blockchain enforced smart contracts. They incorporate methods for managing automated smart contracts which is more efficient and secure matching with hierarchical conditional structures and transfer of contract between various smart contracts. Blockchain enforced smart contracts enables services which are automated, efficient, secure and allow resource distribution. The blockchain technology provides sustainable and efficient method to the existing service structures where some are underperforming and with unreliable security. The algorithms and case-studies proposed in this paper shows the robustness, security, scalability factor and its unique features to provide better-user experience to the user.

VI. REFERENCES

[1] Vitalik Buterin, "A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM", Ethereum White Paper.

[2] Yu Nandar Aung, Thitinan Tantidham, "Review of Ethereum: Smart Home Case Study", 2017 2nd International Conference on Information Technology (INCIT)

[3] Ali Kaan Koç, Emre Yavuz , Umut Can Çabuk , Gökhan DalkÕlÕç , "Towards Secure E-Voting Using Ethereum Blockchain", 2018 IEEE

[4] Thang N. Dinh, My T. Thai, "AI and Blockchain: A Disruptive Integration", IEEE COMPUTER SOCIETY.

[5] Zhongxing Ming∗, Shu Yang†, Qi Li∗, Dan Wang‡, Mingwei Xu∗, Ke Xu∗, Laizhong Cui† , "Blockcloud: A Blockchain−based Service−centric Network Stack", block cloud technical whitepaper.

[6] Christopher Ehmke∗ , Florian Wessling, Christoph M. Friedrich ,"Proof-of-Property – A Lightweight and Scalable Blockchain Protocol", 2018 ACM/IEEE 1st International Workshop on Emerging Trends in Software Engineering for Blockchain

[7] Andrei Sambra ,Amy Guy, Sarven Capadisli, "Building Decentralized Applications for the Social Web", WWW 2016 Companion , April 11–15, 2016, Montréal, Québec, Canada. ACM 978-1-4503-4144-8/16/04. http://dx.doi.org/10.1145/2872518.2891060

[8] Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, "Untangling Blockchain: A Data Processing View of Blockchain Systems", 2017 IEEE.

[9] Massimo Bartoletti, Stefano Lande, Livio Pompianu,Andrean Bracciali, "A general framework for blockchain analytics", 2017 ACM.

[10] Matthias Wichtlhuber, Peter Heise, Bj¨orn Scheurich and David Hausheer, "Reciprocity with Virtual Nodes: Supporting Mobile Peers in Peer-to-Peer Content Distribution", 9th CNSM 2013.

[11] Satoshi Nakamoto (24 May 2009). "Bitcoin: A Peer-to-Peer Electronic Cash System"

[12] Patrick Dai, Neil Mahi, Jordan Earls, Alex Norta, "Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform", 2017 IEEE

[13] Craig wright, Antoaneta Serguieva, "Sustainable Blockchain-Enabled Services: Smart Contracts", 2017 IEEE International Conference on Big Data (BIGDATA)