How to cite this paper:

Rizal Mohd Nor, M.M Hafizur Rahman, Towfiqur Rahman, & Adam Abdullah. (2017). Blockchain sadaqa mechanism for disaster aid crowd funding in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 400-405). Sintok: School of Computing.

BLOCKCHAIN SADAQA MECHANISM FOR DISASTER AID CROWD FUNDING

Rizal Mohd Nor[1], M.M Hafizur Rahman[2], Towfiqur Rahman[3] and Adam Abdullah[4]

[1]*Department of Computer Science,Kulliyyah of ICT, IIUM, Malaysia, rizalmohdnor@iium.edu.my*
[2]*Department of Computer Science,Kulliyyah of ICT, IIUM, Malaysia, hafizur@iium.edu.my*
[3]*Department of Computer Science,Kulliyyah of ICT, IIUM, Malaysia, shommo@gmail.com*
[4]*Institute of Islamic Banking and Finance, IIUM, Malaysia, adamabdullah@iium.edu.my*

ABSTRACT. There exist many online donation platform in the world and yet issues concerning extra fees, accountability and processing delay still exist. In this paper, we propose a decentralized, authentic and transparent donation system to address these issues. We explored several blockchain technologies and formulate smart contracts to be used in this sadaqa system to provide an efficient means to raising funds and managing aid relief for victims during disasters.

Keywords: blockchain, fintech, crowd-funding, etherium, smart contracts

INTRODUCTION

Natural disaster is a natural phenomenon that result in tragic loss of life and cause extensive damage every year. Many victims require help from aid agencies to recover from various types of losses from financial aid to basic necessities such as food and water. This paper discussed the problems of the modern day donation system and ways to improve it. We proposed a decentralized transaction record management system that serves as a platform for donors to donate money or goods to other users who have requested for the donation. All of this occurring securely and with complete trust. Our proposed system lets donors communicate with vendors who could deliver goods to the recipients of the donation and allow users to see all transaction records with complete transparency in all transactions.

Fortunately, humanity has seen a growing trend of charity movements. Many NGOs around the world regulate several billion dollars of donated money every year. Funds are usually received via online donations through bank transfers, Visa or Bitcoin. Unfortunately, modern day transactions still have issues related to transaction fees, potential fraud, absence of accountability, and transaction time. To illustrate how bad these issues are, we can refer to an incident in 2015 when India sent one billion USD to Nepal for an earthquake aid relief. The transaction fees for transferring money to Nepal was one percent. Which means that ten million USD was used for transaction fees that could otherwise be used for relief (Grad, 2016). Another issue is the potential fraud by a middle man regulating and administrating the funds (Davidson, Primavera and Potts, 2016). Consider the earthquake in Haiti in 2010, a recent article notes that the Red Cross raised over $500 million to give to a variety of emergency aid services and build 130,000 homes for the victims. Unfortunately, only six was reportedly built (Grad, 2016). Part of these issues are due to the absence in accountability of

how cash is being spent between the donor, philanthropic organizations, NGOs and civil society organization's (Davies, 2015). Worst of all, the victims usually do not get aid fast enough since it takes a long time to transfer funds from a charity fund to the people who are in need of the donation (Davidson, Primavera and Potts, 2016).

LITERATURE REVIEW

Blockchain

A blockchain is an open distributed database that monitors cash, merchandise traded or transactions on an open decentralized ledger. In a conceptual view, the block-chain is a data structure that consists of time ordered, linked blocks that contain a number of transactions, and each transaction in the public ledger is verified by consensus from a majority of the participants in the system. Once information is entered into the blockchain, it can never be erased (Peter and Panayi, 2015). The blockchain allows trustless network, whereby two strange parties can perform secure electronic transactions without trusting each other (Watanabe et. al., 2016). Crosby, Nachiappan, Pattanayak et al. (2016) concludes that blockchain technology is very attractive and useful to overcome the financial also the non-financial industry dilemma.

Smart Contracts

The concepts of smart contract has been established by Szabo (1997) 20 years ago. Nowadays, the industry has emerged to the second generation of blockchain applications which incorporates smart contract, intellectual property and digitizing asset ownership (Peter and Panayi, 2015). The blockchain smart contracts contains scripts that are stored on the blockchain with a unique address so that it can easily be trace. Juels, Kosba and Shi (2015) notes that decentralized smart contracts has its own advantages compared to the traditional cryptocurrencies like bitcoin. The advantages like fair exchange, to minimized interaction among parties and it is also can enriched transactions.

METHODOLOGY

In order to propose a suitable system for sadaqa based on blockchain, we studied various blockchain platforms and listed its capabilities as shown in Table 1.

Table 1. Comparison of attributes among different blockchain platforms

Blockchain Platform	Consensus Model	Proof Method	Support smart contracts	Permissioned or Permissionless blockchain	Built in Cryptocurrency
NEM (MIJIN & CATAPULT)	Eigen trust	Proof of Stake	No	Permissioned block chain	None
ERIS (FOSS)	Byzantine Fault tolerant	Proof of Work	**Support smart contracts**	Permission less	None
ERIS (MONA)	Byzantine Fault tolerant	Proof of Work	Yes	Permission less	None
Bluemix Hyperledger	PBFT, others can be implemented	Proof of work & Proof of Stake	Yes	Both can be set up	None
Bitcoin	Byzantine Fault tolerant	Proof of Work	No	Permission less	Bitcoin

Ripple	Ripple Consensus Algorithm	Unique Nodes List	No	Permissioned	Ripple(XRP)
Ethereum	Byzantine Fault tolerant	Proof of work	Yes	Permission less	Ether

After studying these blockchain platforms, we concluded that Ethereum blockchain is most suitable for our sadaqa platform because Ethereum can be seen as a transaction based state machine which can transition between states using cryptographically secured transactions (Wood, 2014). When creating a new state machine nodes encode rules or criteria that must be met in order for valid state transition to happen, this information is then merged into blocks and gets uploaded on the blockchain. This functionality of Ethereum allows us to create automated contracts to be enforced between our system actors.

System Actors and Assumptions

For the sake of clarity, we make the some simplifying assumptions before a user can participate in our system. A users must run a full Ethereum node, have an Ethereum account and have an initial Ethereum currency as capital. Additional, we assume there exists a trusted administrator that acts as an initial intermediary between users and the outside world. The administrator handles registration of users to interact with the system.

System actors are defined as *Donors*, *Vendors*, *Recipient* and *Administrator*. A Donor is a user who sends money to other users in the intention to donate. Donors can send money directly to a recipient's workstation or to vendors who deliver goods to the recipient's. A Vendor is a user who holds access to goods and services (i.e. food, clothes, water, electric). Vendors receive money from donors and supplies goods to the recipients. A Recipient is a user who receives money or goods from other users. In general this would be a person with a need for donation, such as victims of flood. Finally, an Administrator is a trusted authority that registers users and transfers money in the system.

System Architecture

Our system empowers Donors to donate money to fundraising organizations or send money for goods (clothing, toys, food), to solicitors who then distributes it to the people who are in need of charity. In our system, the block content stores the Recipients identity, Donors identity, amount of funds to send, and conditions of a contract. A Vendors node can create a smart contract containing the estimated price and quantity of goods to be supplied to the Recipients address, a Donor node can view contracts and appends funds to this Vendor and recipient relationship contract (we use solidity a turing complete contract scripting language to interact within users and external contracts in the Ethereum blockchain network). This lets the Vendor receive an automated notification containing a proof of payment from the Donor node but the Vendor cannot withdraw the funds until the Recipients' receives the goods and then sends a claim message accepting the transaction. Smart Contracts are self-enforcing, and monitor inputs from trusted sources, thus allowing the previously stated Donor and Vendor to acknowledge the claim message send from the Recipient node and finally transfer the funds set in the contract by decreasing the amount of funds from the contract and adding the same amount to the Vendors account. In this stage the methods in the contract clears the total price of the goods(tokens) created by the Vendor, and Recipients' receives some tokens of same value for collecting goods later. This keeps participants informed and engaged in the assessment of their records. We include on the blockchain a cryptographic hash of the transaction record to ensure against tampering, thus guaranteeing data integrity.

Our proposed system prioritize convenience by offering a function that stores a set of reference pointer to all transactions committed by a particular user, which creates a solitary perspective to check all transaction records and be notified for any updates. A syncing algorithm handles communication between the web interface and the back-end server.

IMPLEMENTATION

We have introduced four smart contracts in our proposed system that functions to authenticate users, let users create transaction, keeps reference of each transaction and notifies users about the status of transaction as describe in most implementation by Zyskind and Pentland (2015). An illustration of our implementation is shown in Figure 1. The figure shows how each actor actions is tied to a contract and how these transactions are in then recorded on a blockchain.

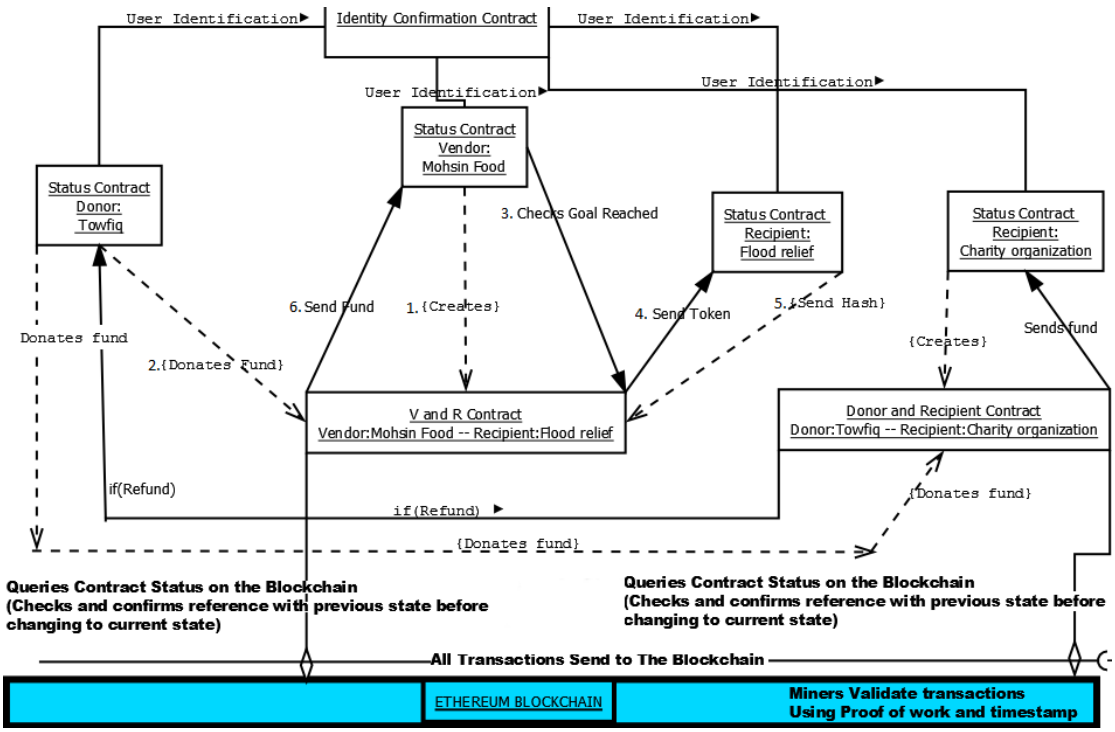

Figure 1. Interactions within smart contracts and blockchain

Identity Confirmation Contract

This public contract is used to register users as Donors, Vendors or Recipients. This requires the user to contact an Administrator and is the prerequisite to every other actions. The administrator manually verifies the users' identity by observing original documents such as an identification card. Our system then stores the ethereum public address, which is a string value referencing the name of the organization, some information about the user or the organization, and an account balance of the user in our systems database. The Administrator sets user permission to act either as a Donor node, Vendor node or Recipient node. We implement different policies in the contract to register users. The roles of actors in this contract is therefore predefined. The contract is then deployed in the blockchain. Policies are in essence a set of permission a user is granted to interact with. The Identity Confirmation

Contract maps the users address on the blockchain and it also keep tracks of a special contract described later, called the Status Contract.

Vendor and recipient agreement contract

This smart contract is built to allow the following users to interact: the recipient which has requested the goods from the vendor, the vendor who supplies the good as tokens, other users who wants to sponsor the vendor to supply the goods to the recipient's we call them donors and Ethereum nodes that are mining the blockchain to verify the transactions. Our contract is created by a Vendor, we have coded methods in this contract which allows a vendor to create tokens by subtracting some value from the Vendors account and use it to set value of the tokens. With functions coded in the contract the Vendor creates funding goal for a charity campaign. Then the vendor nominates the recipient of the token providing the recipients public address in the field necessary. The contract has a method that lets Donors contribute to the contract until it reaches its goal. This is done by mapping the address of all the Donors contributing in the charity. This contract is then uploaded in the Ethereum blockchain to get validated by the miners.

This process validates the member variables of the contract. This creates a block with a current state. Moreover, our proposed system lets users to call the contract's internal functions to read from the contract or change its state by writing to it. Now the Vendor invites other Donors to sponsor the funding goal. The Donor transfers funds to the Vendor by calling a function on the smart contract, where it is held until further confirmation. This information gets uploaded in the blockchain and waits to be mined. A new block is created and gets appended and this changes the current state. Each newly created block is affixed with the hash of the previously created block, to guarantee that data have not been altered at the source (Buterin, 2014). During the lifetime of the contract everyone can see who the recipient is, how much Ether has been raised and from whom (Although the Donors can be anonymous). The system web interface retrieves the contract from the blockchain by mapping their contract address and transaction hash which locates the block state.

Our proposed system lets the Vendor check the total amount of funds received from various Donors. This returns a Boolean value if funding goals is reached and tokens are transferred, otherwise the contract is open for contributions. This function has an if statement which checks if the amount value stored in the contract is equal to the funding goals. If the goals are matched, then the system will transfer the tokens to the recipients address and hold the contributed amount balance as an escrow account (as a third party) until it receives claim message from the Recipient (Roy, 2015). This information gets uploaded in the blockchain, after getting mined by the miners, the recipient receives notification and accepts the tokens after sending a claim message, and this claim message is a hash of a random number known only to him.

Status Contract

This contract works as transaction history for all users interacting in a contract. This contracts stores reference to the Vendor and Recipient agreement contract and Donor and Recipient contract to list down all Donors, the Recipient, the Vendor id and amount of donated balance transferred in the system. Donors will have their Status contract containing pointers to all the Vendor and Recipient it has interacted with, and amount of donated balance it has send in the contract. Likewise Vendors will have their Status Contract containing information like the address of the Recipient and Donor's address or a list of address of all the donor contributing to reach fund goal. And finally the Recipients status contract will store Vendors address it is receiving tokens from, or other Donor's address which communicates

with the recipient node by transferring funds. We built another function to notify users for state changes in the Vendor and Recipient agreement contract mentioned above.

Each transaction from users to the contract stores a status variable. This shows whether "the Vendor has newly created the Vendor and Recipient agreement contract", "is the funding goal reached in the contract", "if new donors have contributed to help fulfill the funding goal", "if the tokens have been sent to the recipients", "if the contract is pending for the recipient to send the hash message" or "if the amount donated value been transferred to the vendor".

CONCLUSION AND FUTURE WORK

To conclude, we have proposed a system of philanthropic donation platform that is distributed, transparent and secure. By storing all transaction details on a public blockchain and by creating smart contracts which interacts with actors within the blockchain system. By doing this we can help donors, vendors and donation receivers from all over the world to transact money in a decentralized, transparent, trusted and secure environment. Furthermore, because the system does not rely on an intermediary to transfer funds, the speed and cost for handling aid is reduced. In the future, we hope to explore on methods that could verify transactions much faster. For example, instead of using proof of work we could experiment with other methods of consensus algorithms such as proof of stake or proof of importance to achieve faster verifications of transactions.

REFERENCES

Crosby, M., Nachiappan, Pattanayak, P., Verma, S., Kalyanaraman, V.: Applied Innovation Review. Applied Innovation Review (2), 6–19 (2016)

G. W. Peters and E. Panayi, "Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money," *arXiv Prepr. arXiv1511.05740*, pp. 1–33, 2015.

G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," *Ethereum Proj. Yellow Pap.*, pp. 1–32, 2014

G. Zyskind and A. S. Pentland, "Decentralizing Privacy : Using Blockchain to Protect Personal Data," 2015.

Jason Grad, "How Blockchain Tech Can Help Solve Problems in Charity | Finance Magnates," 2016. [Online]. Available: http://www.financemagnates.com/cryptocurrency/bloggers/blockchain-tech-can-help-solve-problems-charity/.

Juels, A., Kosba, A., Shi, E.: The Ring of Gyges : Investigating the Future of Criminal Smart Contracts. Online manuscript pp. 1–28 (2015), http://www.initc3.org/files/Gyges.pdf

J. Davidson, Sinclair and De Filippi, Primavera and Potts, "Disrupting Governance: The New Institutional Economics of Distributed Ledger Technology," *Ssrn*, pp. 1–27, 2016.

R. and C. A. F. Davies, "Giving a Bit (coin): Cryptocurrency and philanthropy," 2015.

Szabo, N.: Formalizing and Securing Relationships on Public Networks. First Monday 2(9), 1–21 (1997)

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., Kishigami, J.J.: Blockchain contract: A complete consensus using blockchain. In: 2015 IEEE 4th Global Conference on Consumer Electronics, GCCE 2015. pp. 577–578 (2016

V. Buterin, "A next-generation smart contract and decentralized application platform," *Etherum*, no. January, pp. 1–36, 2014.