Using Distributed Ledger Technology for Shareholder Rights Management

Extended Abstract

Grammateia Kotsialou
King's College London, UK
grammateia.kotsialou@kcl.ac.uk

Luke Riley
King's College London, UK
luke.riley@kcl.ac.uk

Amrita Dhillon
King's College London, UK
amrita.dhillon@kcl.ac.uk

Toktam Mahmoodi
King's College London, UK
toktam.mahmoodi@kcl.ac.uk

Peter McBurney
King's College London, UK
peter.mcburney@kcl.ac.uk

Paul Massey
Crowdcube, UK
paul@crowdcube.com

Richard Pearce
Crowdcube, UK
rich.pearce@crowdcube.com

ABSTRACT

We study and develop an automated platform for shareholder rights management using Distributed Ledger Technology (DLT), in collaboration with an equity crowdfunding company. DLT has high impact potential for the multi-agent systems domain, as it allows participants to agree on the values of shared variables and keep a history of how the values change over time. DLT also enables participants to know that the shared values are common knowledge. In our application, the shared variables that the agents agree on are the shareholder rights. Knowing that there is common agreement on these rights allows us to develop related applications, such as a shareholder voting system. In this paper, we discuss the shareholder rights management platform and briefly mention a related shareholder voting system, both currently under development.

KEYWORDS

Distributed Ledger Technology; Blockchain; Shareholder Rights

ACM Reference Format:
Grammateia Kotsialou, Luke Riley, Amrita Dhillon, Toktam Mahmoodi, Peter McBurney, Paul Massey, and Richard Pearce. 2018. Using Distributed Ledger Technology for Shareholder Rights Management. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Distributed Ledger Technology (DLT) is important for multi-agent systems as it is a new method for a decentralised network of agents to maintain consensus on data, even if they may not trust each other. In this paper, we focus on blockchains, which are a type of DLT. A blockchain is a series of blocks of data linked together (via cryptographic hashes), which increases in length over time as new blocks keep being added. To add a block in the chain, the new data received has to be validated by a large enough number of validators (a class of agents). The rules of how blocks are added and what data are allowed within a block are described by the consensus protocol [1]. Once a block is added, it is fixed (cannot be modified) and everyone in the system has a copy of the entire updated blockchain. Thus, the data in a blockchain is essentially immutable[1].

Blockchains can be categorised [3] as follows: (*a*) *Public (permissionless)* - where any agent can download, read and write to a blockchain; (*b*) *Private (permissioned)* - where the download, read and write capabilities are assigned by a pre-selected group of agents, e.g. multiple institutions in the same sector.

In corporate governance, DLT can have a revolutionary impact [4] due to the following: greater transparency of company ownership; easier administration; and creation of an infrastructure on which to build innovative applications, e.g. real time accountancy, where business transactions would be posted in real time to a blockchain. For private companies, corporate governance includes managing the company shareholders' rights and privileges, e.g.: priority rights to compensation in the event of a bankruptcy of the company; voting privileges; and pre-emption rights (where some shareholders can have a priority on buying shares over other potential investors). At present, shareholders' rights are generally recorded in written contracts and the relevant information is held only in spreadsheets. These records may be kept by an independent third-party, such as a law firm, and are often hard to access. Thus shareholders keep their own records.

By detailing the shareholder rights as shared variables on a blockchain, we can greatly reduce the administration burden of manually maintaining all the related documentation. Another advantageous feature of this application is the ability to automatically trigger some actions for a shareholder should certain pre-conditions occur. This can be achieved by placing self-executing scripts on the blockchain, known as *smart contracts.* Finally, given the common knowledge base that DLT offers, we can build additional applications on the blockchain such as a shareholder voting system.

In this work, we discuss the theoretical model of maintaining shareholder rights on a blockchain and describe an example that focuses on a shareholder voting system. The role of agents within our application is to: (i) validate that only correct data can be added

[1]This statement assumes there are enough validators in the system to make this a near impossible task.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

to the blockchain; and (ii) perform reads and attempt writes to the common knowledge base of the blockchain.

2 THE MODEL

We define two categories of agents: the *users*, such as the shareholders of a company, associated lawyers and the equity crowdfunding platform raising the funds, and the *validators*, who validate changes to the shared variables and add them to the distributed ledger. We use U and V for the sets of users and validators, respectively.

States. A state P represents the common knowledge of the data relating to the company, e.g., the value that each shared variable is currently assigned. Let $\mathcal{P}$ be the set of all possible states P.

Actions. Given the occurrence of specific pre-conditions, an action is a change of state that a user has the legal authority to perform according to the company's documentation. Therefore, we can think of actions as the ability to change some of the shared variables given certain circumstances. For example, if the shared variable $SellingSharesAllowed = true$ then a shareholder can choose to perform the action $ListMySharesToSell$, which will increase the shared variable $NumberOfSharesBeingSold$. Additionally, smart contracts can also be placed by a user to set actions to automatically trigger every time the pre-conditions are met. For instance, a smart contract could be made to trigger the action $ListMySharesToSell$ in every state where the shared variable $CurrentSharePrice > 100$.

Shareholder rights. These are the collection of legal actions that a shareholder can perform given the current state. Let A define the set of all possible actions. Given a state $P \subseteq \mathcal{P}$, the set of rights that each user $i \in U$ has is defined by $R(P, i)$ with $R(P, i) \subseteq 2^A$.

Block. We define as a block b a collection of data which has been confirmed by the validators. For this application, the data in a block that we are interested in are as follows: (i) the modified shared variables (so we can update the current state in order to keep an accurate view of each shareholder's current rights); (ii) what actions have just been performed by users (to keep a legal audit trail); and (iii) what smart contracts have been added (to predict future states).

Blockchain. A blockchain C^l is defined as a sequence of cryptographically interlinked blocks $[b_1, b_2 \ldots, b_l]$, where a block $b_t \in C^l$ is linked only to the previous block $b_{t-1} \in C^l$. The length of the sequence increases over time as new blocks are added (new data is validated). Note that by adding a block to the chain, the common knowledge is updated. This is captured by the function $S : C^l \to P$.

A user i can attempt to change a shared variable listed on the blockchain by sending an action $\alpha \in A$ to the validators. We take the benevolent assumption for the validating agents, which implies that a new action α, sent by an $i \in U$, will be added to the blockchain C^l iff $\alpha \in R(S(C^l), i)$.

2.1 Initial Implementation

Our implementation focuses on shareholder rights related to voting. For privacy reasons, there is a new blockchain created for every company using our system, and the users of each blockchain are only the company's stakeholders, e.g. shareholders or associated lawyers. We use a private blockchain architecture where our equity crowdfunding partner is the only trusted organisation that can create new blockchains (one for each new company using this platform). Before a blockchain is created, the shareholders of the related company need to digitally sign a (Ricardian) contract representation of a simplified version of the company's shareholders' agreement document, e.g. the Articles of Association document. A Ricardian contract provides a legal document together with its related computer code [2], therefore signing this contract indicates that the shareholders agree to the running of this document's code on the blockchain. Then the shared variables of the blockchain are initialised to the values agreed in the signed Ricardian contract. The hash of the digitally signed Ricardian contract is placed in the first block of the chain for audit reasons. The users of the blockchain can now modify the shared variables according to their rights originally defined in the shareholders' agreement document of the company.

2.2 An Example

Consider a company and the set of its shareholders: $\{1, 2, 3\} \in U$. Agents 1 and 2 have each a 40% of the shares while agent 3 has 20%. Agents 1 and 2 have shares with voting rights (including starting a voting process) and agent 3 has shares without voting rights.

Assume that this information is written in a Ricardian contract digitally signed by all shareholders. Then blockchain C^1 is created for the company, where the shared variables are: $SharePercentage$, $VotingRights$, $VoteQuestion$ and $VoteOptions$. The block b_1 initiating the chain includes the Ricardian contract code that contains: (i) the initial values of the shared variables and (ii) the set of allowable actions. The shared variables are initialised as follows: $SharePercentage = \langle 40, 40, 20\rangle$; $VotingRights = \langle T, T, F\rangle$; $VoteQuestion =$' '; and $VoteOptions = \langle\rangle$. From the shared variables, we can infer that the current state is given by $P_1 = [\langle 40, 40, 20\rangle, \langle T, T, F\rangle, \text{' '}, \langle\rangle]$. The set of available actions in this example is: $A = \{startAVote(question, options), vote(option), closeAVote(question)\}$.

Recall that, given a state, the R function outputs the rights of an agent. For example, the rights of agent 1 are given by $R(P_1, 1) = \{startAVote(question, options)\}$. Observe that, given state P_1, agent 1 can only start a vote. She cannot cast a vote or close a vote due to the fact that there is no ongoing voting process, indicated by $VoteQuestion =$' '. For agent 3, we get that $R(P_1, 3) = \emptyset$ implying that she currently cannot perform any actions. This is indicated by the initial shared variable value $VotingRights[3] = F$.

Let agent 1 perform the action $startAVote$('*John Smith to be appointed CEO*?', $\langle yes, no\rangle$). Then this action is confirmed by the validators, added to a new block b_2 (interlinked to b_1), and the blockchain is updated to C^2. The current state is now $P_2 = [\langle 40, 40, 20\rangle, \langle T, T, F\rangle$, '*John Smith to be appointed CEO*?', $\langle yes, no\rangle]$. Observe that the rights of agent 1 change to $R(P_2, 1) = \{vote(\{yes, no\})\}$, which is indicated by $VotingRights[1] = T$ and $VoteQuestion \neq$' '.

This process of shareholders performing legal actions, which change the common knowledge of the system, can continue indefinitely, with the blockchain C recording a history of the shared variables and the actions that has occurred.

3 CONCLUSIONS AND FUTURE WORK

In this short paper, we described a shareholder rights management platform on a blockchain, with a voting system example. Future work will focus on expanding the implementation to include complex voting systems with a large list of shareholder rights to be represented. Of our special interest is a game-theoretical approach to this model, where we allow self-interested validators in the blockchain (dropping the benevolent assumption). Such agents may attempt to ignore shareholder actions that they do not agree with.

REFERENCES

[1] Arati Baliga. 2017. Understanding Blockchain Consensus Models. *Persistent Whitepaper*. https://doi.org/wp-content/uploads/2017/04/WP-Understanding-Blockchain-Consensus-Models.pdf

[2] Vinay Gupta, Rob Knight, Aeron Buchanan, Christopher Wray, Ian Grigg, Casey Kuhlman, Michael Mainelli, and Clive Freedman. 2017. Smart Contracts. Real Property. *Mattereum Whitepaper*. https://doi.org/assets/MattereumDraftforPublicComment.pdf

[3] The Alan Turing Institute. 2017. Insight Report on Distributed Ledger Technologies. *Report Series: No. 2017.4*.

[4] David Yermack. 2017. Corporate Governance and Blockchains. *Review of Finance, Volume 21, Issue 1*.