Deploying a Shareholder Rights Management System onto a Distributed Ledger

Demonstration

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

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

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

ABSTRACT

This work demonstrates how a multi-company shareholder rights management system has been implemented using Distributed Ledger Technology (DLT). In this demo[1], we use a permissioned blockchain to store our corporate data, such as the list of all registered companies, each company's shareholders and how many shares everyone holds. It is assumed that the nodes of the blockchain are controlled by the main stakeholder agents but we show that users who do not run a node can still use multiple websites to access company information. On top of this, we show our system can be used to allow any shareholder to participate in elections for company matters. Lastly, we describe how we designed our system's architecture so that it could be implemented even on a public blockchain.

KEYWORDS

Distributed Ledger Technology; Blockchain; Shareholder Rights Management

ACM Reference Format:
Luke Riley, Grammateia Kotsialou, Amrita Dhillon, Toktam Mahmoodi, Peter McBurney, and Richard Pearce. 2019. Deploying a Shareholder Rights Management System onto a Distributed Ledger. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Developing protocols for self-interested agents to come to agreement on what data is saved, shared and valid can be a challenging aspect of multi-agent systems. *Distributed Ledger Technology* (DLT) [1] provides a unique solution to this problem through its *consensus protocols* that allow a possibly unlimited and anonymous number of self-interested agents (each running a node of the DLT) to maintain consensus on data without a central authority. A consensus protocol consists of two parts: (1) a *sybil control mechanism*, which give no advantage to agents who create multiple accounts; and (2) a *data agreement protocol* that details the rules regarding how the next valid block of data is agreed.

For this demonstration, we focus on a special type of DLT, called blockchains [11]. A blockchain is a series of blocks of data (containing multiple transactions) linked together via cryptographic hashes. A blockchain increases in size over time as new blocks keep being added. Each node of the blockchain network maintains its own copy of the blockchain. When a node receives a new block, the data within it needs to be validated according to the data agreement protocol. If a node deems a block valid, then the block is added to that node's copy of the blockchain. Therefore, everyone who has the same copy of the blockchain has agreed to all the data within it.

The first blockchain implementation, Bitcoin [11], allows agents to come to a consensus only on data regarding how much Bitcoin everyone has. Since the introduction of Ethereum [5], blockchains can allow agents to reach consensus on any type of data. Ethereum provided this breakthrough via the introduction of *smart contracts*, which are user defined, tamper-resistant and deterministic pieces of code [3, 9]. When saved onto a blockchain, a smart contract is given a unique lookup address so that users can interact with it. Smart contracts can allow data storage of various system defined types (e.g. integers, strings, etc) or user defined types (similar to classes in object orientated programming), albeit with a size limitation of whatever can fit into a block. A smart contract is activated when a user sends a transaction to its unique address that includes metadata on what specific smart contract function to run and with what inputs. This transaction triggers every node of the network to run the corresponding deterministic function code with the same inputs. In this way, blockchains can provides a method to coordinate large decentralised systems of agents (if each agent runs a node).

This demonstration program builds on the previous work [8]. The idea is to use a blockchain to save data related to corporate governance (i.e. the type and number of shares held) and then use this data to coordinate agents on related activities (i.e. executing shareholder votes and performing share transfers). DLT can revolutionise the corporate governance domain on multiple levels [13]. Apart from direct applications such as reducing the burden of administration, DLT can provide the infrastructure for additional desirable features to be built, such as allowing all stakeholders to have transparency on: a company's ownership, shareholder voting privileges and real-time time business transactions. This transaparency could allow any stakeholder to deploy a smart contract on to the DLT as a reactive, social agent[2] to act on their behalf.

[1] See https://goo.gl/4hsSCZ

[2] DLTs cannot currently deploy smart contracts as pro-active learning agents.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

2 DOMAIN DESCRIPTION

Some companies (e.g. crowdfunding platforms such as CrowdCube[3]) have the legal authority to maintain shareholder information and perform actions on behalf of the company (i.e. run shareholder votes). But this information may not be easily shared with relevant company stakeholders such as the shareholders themselves or public company information services (e.g. Companies House[4]).

To allow for transparency on the information held, we have prototyped a system where the company information does not only exist on paper (or a single server) but also on the blockchain. This allows the agents running a node to access the agreed valid data and coordinate actions regarding company information (e.g. allowing shareholders to vote when an election begins). Additionally, stakeholders who do not run a node (due to not having the technical expertise or the capital required), can access the company information from multiple access points (should each node provide one, e.g. via a website), which in turn increases the trust on the accuracy of the company's information.

3 ARCHITECTURE

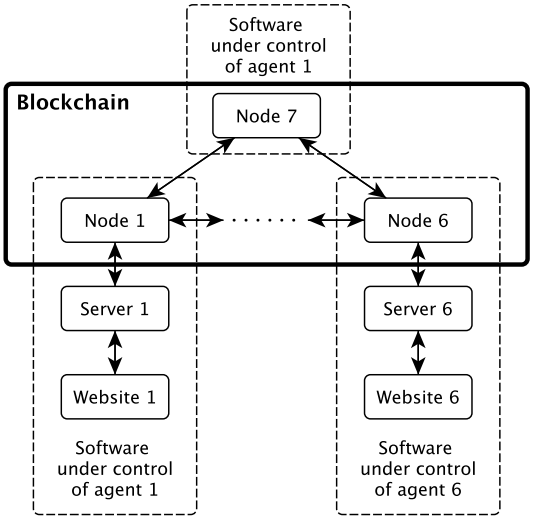

Figure 1: The demo includes seven agents running nodes on a blockchain. Agents 1 to 6 provide access to the company information to other users via a website and server connected to their node. The nodes can be interconnected in any way, not only in a circle as this graph implies.

The main advantage of the architecture we developed is that the supporting system can be implemented using either a *public*[5] or a *permissioned*[6] blockchain. To achieve this: (*a*) we have written our smart contracts in the Solidity language [6] which can be understood by a variety of public blockchains such as Ethereum and Ethereum Classic or permissioned blockchains such as JP Morgan Chase's Quorum [2] and Monax's Hypeledger Burrow [10]; and (*b*) we have developed our website and server code (that interact with the blockchain), using the Truffle development environment [4] and the Web3 javascript package [7], both of which are understood by the Ethereum and Quorum blockchains.

Our demonstration program is based on a seven node permissioned blockchain network deployed on the Quorum blockchain using the *Istanbul Byzantine Fault Tolerance* (IBFT) consensus protocol [12]. IBFT allows new blocks to be produced in a round robin format between the nodes of the blockchain with the relevant permissions. When a new block is produced, the permissioned nodes vote on whether to accept this block into the chain. If more than two thirds of the nodes vote "accept", then the block is added and the decision is final. More specifically, the IBFT protocol can withstand up to $m = \frac{N-1}{3}$ malicious nodes, where N the total number of nodes. In our demo example, six of the seven agents running a node provide a website for other users (who are not one of the permissioned nodes) to view the company information stored on the blockchain, see Figure 1. Note that, given a user knows the total number of nodes N, if $2 \cdot m + 1 = 2 \cdot \frac{N-1}{3} + 1 = 5$ of the websites display the same information, then she also knows that this is the information written on the underlying blockchain.

To fully deploy this system in practice, it would require more nodes to reduce the possibility of malicious agents forming a coalition and taking over the network. If malicious agents hold a majority on the blockchain, this would have a large negative impact on the network as they could insert erroneous data. However, this situation could be solved externally to the blockchain. As soon as trustworthy nodes discover malicious interference, they can collaborate off the chain together with other stakeholders motivated to recover the system's truthfulness. For instance, they can work out what was the last valid block and then create a new more decentralised permissioned blockchain starting from this block but excluding the malicious agents as nodes. Being excluded from the role of a permissioned node could seriously damage the reputation of the agent (stakeholder) responsible for that node.

3.1 Interfaces

As the agents are separate individuals, they may allow their users to access different features regarding the blockchain data. For instance, in our demo we show how agent 1 allows users to import company data to be added onto the blockchain, while agent 2 does not allow this feature. Whereas both agents provide the following user facilities: displaying what companies have data saved on the blockchain; displaying the shareholder list of the different companies; providing the ability to create a shareholder vote; providing the ability to cast a vote in relevant elections; and providing the ability to view information on the ongoing votes.

Now that users can choose multiple ways to access data recorded about their company and they have the ability to check if the data displayed is correct (by moving to other websites connected to the blockchain), essentially a market is being created between data suppliers (the agents running a node) and data consumers (the users). If the data consumers are rational, they will choose their desired data supplier through a combination of trust and reputation, which is another incentivisation mechanism for the data suppliers to not malicious present erroneous data.

[3] https://www.crowdcube.com/
[4] https://beta.companieshouse.gov.uk
[5] Public blockchains allow any agent to download, read and write to the blockchain.
[6] Permissioned blockchains allow the download, read and write capabilities to be assigned by a pre-selected group of agents.

REFERENCES

[1] Andreas M. Antonopoulos. 2014. *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly.

[2] JP Morgan Chase. 2016 - onwards. Quorum Overview. (2016 - onwards). <https://github.com/jpmorganchase/quorum/wiki/Quorum-Overview>.

[3] Giovanni Ciatto, Stefano Mariani, and Andrea Omicini. 2018. Blockchain for Trustworthy Coordination: A First Study with LINDA and Ethereum. In *IEEE/WIC/ACM International Conference on Web Intelligence, WI*. 696–703.

[4] Consensys. 2017 - onwards. Ethereum Javascript API. (2017 - onwards). <https://github.com/trufflesuite/truffle>.

[5] The Ethereum Foundation. 2014 - onwards. A Next-Generation Smart Contract and Decentralized Application Platform. (2014 - onwards). <https://github.com/ethereum/wiki/wiki/White-Paper>.

[6] The Ethereum Foundation. 2014 - onwards. The Solidity Contract-Oriented Programming Language. (2014 - onwards). <https://github.com/trufflesuite/truffle>.

[7] The Ethereum Foundation. 2015 - onwards. Ethereum Javascript API. (2015 - onwards). <https://github.com/ethereum/web3.js/>.

[8] 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 *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 1986–1988.

[9] Daniele Magazzeni, Peter McBurney, and William Nash. 2017. Validation and Verification of Smart Contracts: A Research Agenda. *IEEE Computer* 50, 9 (2017), 50–57.

[10] Monax. 2016 - onwards. Hyperledger Burrow. (2016 - onwards). <https://github.com/hyperledger/burrow>.

[11] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008). <http://bitcoin.org/bitcoin.pdf>.

[12] PegaSys. 2018. Scaling Consensus for Enterprise: Explaining the IBFT Algorithm. (2018). <https://media.consensys.net/scaling-consensus-for-enterprise-explaining-the-ibft-algorithm-ba86182ea668>.

[13] David Yermack. 2017. Corporate Governance and Blockchains. *Review of Finance* 21, 1 (2017).