Decentralised Autonomous Organisations for Public Procurement

Félix Monteiro
felix.saraiva@tecnico.ulisboa.pt
INESC-ID, Instituto Superior Técnico, Univ. Lisboa
Lisboa, Portugal

Miguel Correia
miguel.p.correia@tecnico.ulisboa.pt
INESC-ID, Instituto Superior Técnico, Univ. Lisboa
Lisboa, Portugal

ABSTRACT

Blockchain has been recognised as a technological breakthrough with the ability to support new decentralised security-based solutions in sectors such as information technology and finance. Blockchain allows different communities to create Decentralised Autonomous Organisations (DAOs), which are self-organised democratic organisations controlled by smart contracts. This paper presents a new DAO model for the procurement of services by public organisations, such as government agencies. To demonstrate the advantages of this solution, this work looks specifically at current public procurement systems that resort to third-party contractors that manage these negotiations. Third parties lack the transparency, security, and democratic representation that a DAO can provide. We present the implementation of a DAO as a set of smart contracts executed on Ethereum-compatible permissionless blockchains, supported by a consensus algorithm, replacing third-party contractors.

CCS CONCEPTS

• **Computer systems organization** → **Dependable and fault-tolerant systems and networks**.

KEYWORDS

Blockchain, Public Procurement, Decentralized Autonomous Organizations, Smart Contracts, Ethereum

ACM Reference Format:
Félix Monteiro and Miguel Correia. 2023. Decentralised Autonomous Organisations for Public Procurement. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE '23), June 14–16, 2023, Oulu, Finland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3593434.3593519

1 INTRODUCTION

From its early stages, blockchain technology has been used to try to solve several problems in our society related to transparency, security, and decentralisation [17]. The first blockchain appeared as part of Bitcoin, leading to profound changes in our economic system and the use of money in the digital world [15]. More recently, practitioners and researchers have started exploring the advantages of developing decentralised web applications having a blockchain back-end running smart contracts, with the expectation that these projects will be recognised by the market and shape the way web applications are developed into a futuristic Web3.

Blockchain technology supports important properties such as decentralisation and automation. These qualities allowed the creation of Decentralised Autonomous Organisations (DAOs) [7, 8, 21, 23]. DAOs are collectively owned and managed organisations that are implemented using a blockchain system. The essence of a DAO is not only its code but also its community. It requires quorum approval to execute tasks or take democratic decisions. Therefore, these organisations do not have a central point of authority, such as a CEO, who might abuse his decision power[4].

In *public procurement*, public institutions aim to contract services from private entities. To achieve that, *contests* are created to determine the most suitable candidate to provide the required service. These contests should be as transparent as possible. However, when contracting is established between two entities with a conflict of interest or when the agent's characteristics are not known by the principal, we have a problem of optimal contracting, known as *Principal-Agent Problem* [6]. In the business world, circumstances often occur when an agent is hired by another entity, the principal, to make decisions on behalf of the principal's interests. However, the agent can decide to act in his own best interest, which can be contrary to the principal's. This situation occurs due to a conflict of priorities and *asymmetric information*, with the agent retaining more information about the assignment than the principal. Therefore, the principal cannot trust the agent to work in its best interest. Introducing a trusted third party might solve this problem if we assumed that this third party had both the principal's and the agent's best interests in consideration. However, this is more of a problem-shifting approach than a problem-solving approach.

The standard model for *public procurement contests* starts with the public organisation publishing a call with a list of responsibilities and procedures that the private entities must follow. This document presents the rules and bylaws of the contest [14], explaining how applicants can submit their proposals, the contest deadlines and the service requirements. Some of these rules may not be fair to all the applicants, and thanks to the applied centralised solution, they have no voice in the matter.

A DAO can be useful in this context because it is non-malicious or manipulative and can guarantee trust as long as its members trust the deployment code. This makes DAOs an excellent solution to the Principal-Agent Problem and a significant improvement in public procurement contests. Furthermore, in many cases, such public organisations enhance bargaining power, increasing the unfairness of such negotiations. The proposed approach completely eliminates two-way negotiations, as it has a linear proposal and a voting process.

EASE '23, June 14–16, 2023, Oulu, Finland

ACM ISBN 979-8-4007-0044-6/23/06.
https://doi.org/10.1145/3593434.3593519

Therefore, this paper proposes that a DAO replaces trusted third parties in these business interactions, acting as the middleman between public institutions and private applicants. In this approach, the applicants and the public organisation communicate directly through the DAO, giving the applicants certainty that the decision made by the public institution is democratic and allows them to verify the integrity of the proposals other applicants submit. Such a system inherits characteristics of blockchain technology, such as decentralisation, security, immutability, traceability, transparency, privacy, and interoperability.

The proposed solution is a DAO focused on being the connection between public institutions and the applicants of a public procurement contest. This solution has three types of members: board members, applicants, and senior applicants. Board members must submit proposals to add new members to the DAO or create new procurement contests. Applicants can propose services to specific contests, and senior applicants can propose changes to the bylaws of such contests. Proposals are decided by the DAO's board, where each member can vote. The presented system is meant to be applicable to any scope of public procurement contests; however, the pre-development research of this solution focused on specific public hospitals' procurement contests. The implementation of the solution is written in Solity and can be deployed on Ethereum-compatible blockchains.

The implementation was tested in an Ethereum testnet regarding cost and performance. The deployment results showed the system's total cost of approximately 0.27 ETH (301,78€ on 30/12/2022), with a total size of 49.908 KiB in EVM bytecode. The performance results showed an average latency of 17.76s and an average cost of 0.004060 ETH per operation.

2 BACKGROUND AND RELATED WORK

This section introduces blockchain technology, smart contracts, Decentralised Applications, and DAOs.

A *blockchain* is a distributed ledger that consists of a decentralised, shared, replicated, and synchronised record of transactions secured by cryptographic protocols [22]. In such a distributed system, there is no trust among the nodes, so transactions are verified before they are coupled and registered in a chain of blocks. Blockchains can be *permissionless*, where anyone can access the ledger, issue a transaction, run a node, or publish a smart contract, or *permissioned*, where nodes need permission to access the network [17].

Smart contracts are computer programs stored and executed in a blockchain. Smart contracts have their own state and have functions/methods that are executed when they are called by special transactions that take input parameters. In the Ethereum blockchain, smart contracts are immutable computer programs that run deterministically in an Ethereum Virtual Machine [3]. They are said to be immutable, as they cannot be changed or updated like regular software; the only way to change a smart contract is to deploy a new, independent instance. They are deterministic because the outcome of the execution will always be the same regardless of the user.

With these technologies, it is possible to build *Decentralised Applications* (DApps). The front-end of a DApp is standard, e.g., a set of web pages or a mobile App, but its back-end is composed of smart contracts [3]. DApps should always be open source to ensure they can be trusted, similar to what happens with blockchain code. At least the back-end does not have a single point of failure, as it is executed on a blockchain and is therefore decentralised [5].

A DAO is a democratic self-organised organisation, without centralised control, instead controlled by smart contracts running in a blockchain [4, 7, 21]. Like DApps, the creation of a DAO is highly based on smart contracts deployed on the blockchain. When the DAO is created and deployed, a poll is the only way to alter its regulations. After its launch, some DAOs can start collecting or distributing funds for the purpose for which it was created. These funds come in the form of *tokens*, which are blockchain-based abstractions that can be owned. Once members participate in the DAO, they can earn rights such as voting in decision polls, providing feedback, or even setting future ideas for the organisation [7, 23]. It is also possible to distribute non-transferable tokens based on a member's reputation in the DAO. Reputation can be earned and lost based on the consequences of a member's actions.

The decision-making process is one of the core features that separates a DAO from standard organisations. Decisions are recorded on-chain and executed automatically. Once a proposal pool obtains a successful decision, that decision is executed without human interference, providing a secure and democratic interaction to end users [20].

Due to the innovation that smart contracts introduced to blockchain technology, it is possible to assemble DAOs to achieve multiple business goals. Protocol DAOs provide decentralised financial services (DeFi); an example is Uniswap, which supports the trading of tokens [13]. Social DAOs focus on supporting people's interaction; an example is the FWB DAO [1].

To our knowledge, there are no current proposals for DAOs for public procurement services.

2.1 Objectives

The objectives of the proposed design are to improve the public procurement process and introduce a new DAO model suitable for public procurement contests. This system aims to address the current challenges faced by the public procurement system, including the participation of third-party contractors and the need for greater traceability and transparency in the negotiation process.

The proposed DAO model eliminates the need for third-party contractors by allowing direct negotiations between the relevant parties. This reduces the cost of the procurement process and improves its efficiency by eliminating intermediaries. The use of blockchain technology will also introduce decentralisation into the negotiation process, allowing an equal distribution of power and decision making.

Furthermore, the traceability and transparency introduced by blockchain technology will allow greater public verification of procurement contracts. This will improve the accountability of the parties involved and increase public trust in the procurement process. Furthermore, automation of the negotiation process will reduce the likelihood of human error and improve the efficiency of the procurement process.

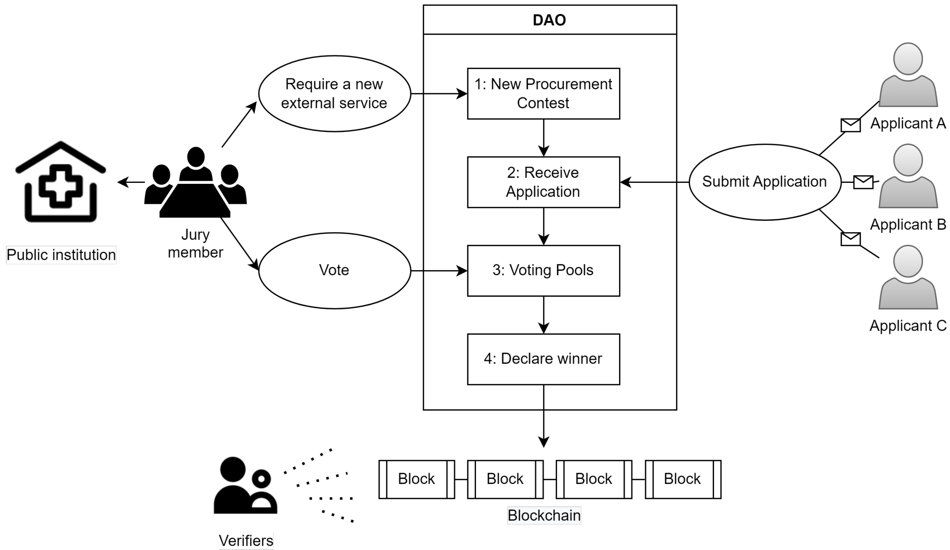

Figure 1: DAO stakeholders interacting in a public procurement contest

2.2 Stakeholders

One of the main characteristics of a DAO is its community and the stakeholders that compose it. Figure 1 shows these stakeholders by representing their interactions in the context of a public procurement contest intermediated by a DAO. There are five types of stakeholders:

(1) *Public institution:* it is one of the DAO's main stakeholders and plays the principal role in the Principal-Agent Problem. Provides a group of contest jury members who represent the public institution on the DAO Board. Responsible for requesting a service from the public, this stakeholder can create proposals and vote on those proposals representing the decision-making entities within the DAO. They are also responsible for adding new members to the DAO and choosing the winners of the contests.
(2) *Applicants:* a group with limited actions within the DAO. When a corporation desires to provide a service to fulfil a request published by the DAO, applicants can submit a proposal with their service details and cost. These stakeholders do not have a vote on the major decisions of the DAO. They benefit from it as an unbiased, transparent, immutable, and secure bridge between them and the public institution.
(3) *Senior Applicants:* applicants that have participated in several contests and won at least w (e.g., $w = 3$) are considered to have won a fair amount of reputation within the DAO's community to be upgraded to Senior. They represent an applicant with the power to present feedback proposals to be considered by the public institution.
(4) *Verifiers:* in the DAO, not only members can verify the transactions and decisions made. Anyone can access this information publicly for legal or other purposes.
(5) *Public Procurement DAO:* the DAO itself represents a stakeholder since it takes autonomous decisions like upgrading a regular applicant to senior, declaring the winner of a contest, removing a member, or executing other actions depending on a proposal's decision, with the aspect of storing those decisions on the blockchain.

2.3 Governance

Every DAO is built around a governance protocol that manages how decisions are made and how the DAO community interacts to make decisions.

This DAO's governance core comprises two contracts, `TimeLock` and `GovernanceProtocol`. The `TimeLock` contract uses the module `TimeLockController` [16]. When set as the owner of an `Ownable` smart contract, the `TimeLock` contract enforces a timelock on all `onlyOwner` operations. A *timelock* is a mechanism that delays calls to another smart contract until a predetermined time has passed. This makes `TimeLock` the only executor of operations in this system and provides the community with time to process the result of an operation before its next stage.

The `TimeLock` contract has three important system roles: *Proposer*, *Executor* and *Admin*. The admin role is automatically assigned to the deployer, allowing him to assign roles. The Proposer role is given to the `GovernanceProtocol`, making it in charge of proposing operations and the only entity allowed to schedule and cancel operations. The Executor role is given to everyone, meaning anyone can execute the operations previously scheduled by the Proposer. After deployment, the Admin role is renounced in favour of administration through time-locked proposals. Without administrators, it is impossible to change roles later, making this system fully autonomous. This relationship between the `GovernanceProtocol` and the `TimeLock` contracts makes this project a decentralised and autonomous system.

The `GovernanceProtocol` contains all the functionality logic connected to the user interface, making it the direct point of interaction for the system users [16]. Through the `GovernanceProtocol` contract, users can submit all available operations that the `TimeLock`

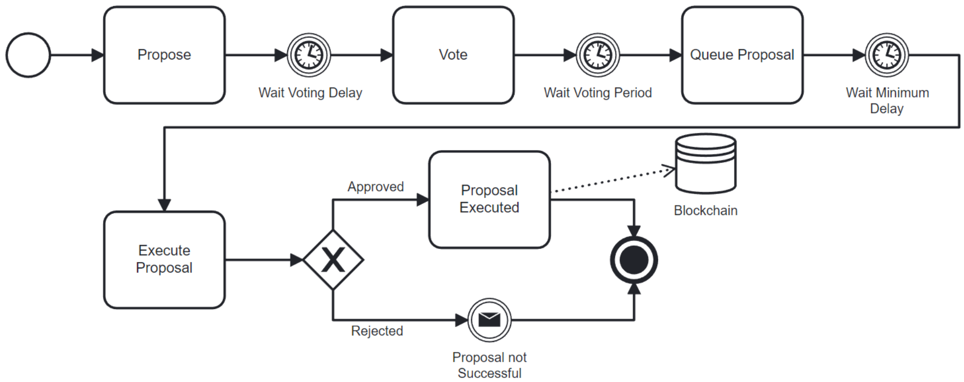

Figure 2: Public Procurement DAO Activity UML Diagram

schedules. These include propose, vote, queue, and execute proposals.

Although this system has role verification for voters, the governance protocol identifies voters, not by their role, but by checking if they hold a certain *token*: the *Governance Token* (GT) is a standard ERC20 token modified to apply on-chain voting. ERC20 is a standard for fungible tokens, i.e., for tokens that are identical and all have the same value [9]. In this design, a board member must have at least one GT token to vote. However, the token balance of each board member does not account for voting power, making token transfers cheaper. Only board members can mint and delegate tokens to themselves since they are the only voting authorities in the DAO.

3 PUBLIC PROCUREMENT DAO

The proposed design is a Solidity-based DAO to be deployed on a permissionless blockchain, focused on being the connection between public institutions and the applicants of a public procurement contest. This system follows a set of requirements to allow governmental institutions to see its advantages compared to the approaches currently used, mentioned in Section 1. The remaining contracts in this system are shown in Figure 3 and are not related to governance. They implement the actual functionality with which proposals can interact if successful.

3.1 Functionality

This subsection presents the system functionality from the user's perspective and links that functionality to the goals of the DAO. The system operates in a proposal-based interaction, maintaining all its functionalities democratic. This proposal-based workflow can be seen in Figure 2. The workflow of a proposal starts with its submission, requiring the proposing user's wallet address and private key to sign the transaction. The user shall select the proposed DAO's functionality and introduce the required function parameters.

The DAO offers different features per member type with different types of proposals:

- *Board members* can mint GT tokens, propose a new contest, propose to add/remove board members, propose to add/remove suppliers, and vote on proposals;
- *Applicants* can submit contest applications and propose to add/remove senior suppliers;
- *Senior applicants* can do the same as regular applicants and submit feedback proposals.

After a proposal is submitted and before its voting period begins, the system enforces a configurable voting delay, allowing board members to prepare for the next stage and mint last-minute governance tokens if necessary. After this delay, the voting power is fixed, and the voting period starts. If a board member's account already owns GT tokens, it is not required to mint again for the next vote. During the voting period, each board member can submit one vote by providing the respective proposal ID, a voting option, and a vote description that justifies the reason for the voting option. The voting options can be "For", "Against", or "Abstain".

Governance Tokens are not spent per vote, so one token is enough to prove to the governance protocol that a board member can have one vote in multiple proposals. Consequently, if a board member submits a vote without holding a governance token, that vote is not considered.

When the voting period ends, the majority wins, meaning that even if a board member with 20 GT votes in favour and two board members with 1 GT each vote against, the result of the proposal is unsuccessful. After the voting period, the proposal is ready for queueing.

The queue period is set according to the minimum delay time set by the timelock. It is employed so that the DAO's community can prepare for future changes in the upcoming execution. The queue and execute functions require passing all proposal parameters, instead of just the proposal ID, because this data is not stored on-chain, as a measure to save gas. However, these parameters can be found in the events emitted by the contract. The only parameter not sent in full is the proposal's description since it is only needed in its hashed form to compute the proposal ID.

After the minimum delay time, the proposal can proceed to execution. From the user's point of view, executing a proposal is the same as queuing it. In a successful proposal, the proposed function is executed with the parameters provided. The system presents the error: "Proposal not successful" in a failed proposal.

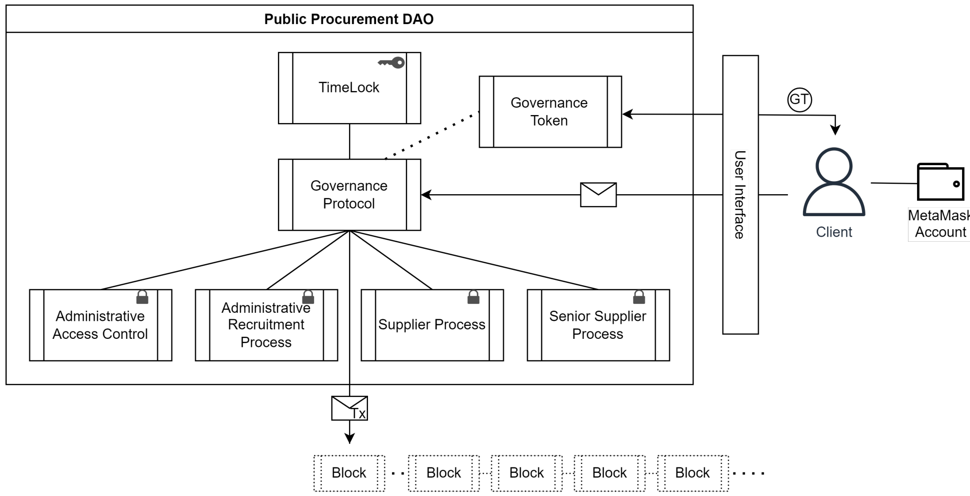

Figure 3: Public Procurement DAO high level architecture

In essence, system functionalities are controlled by the collective decision of the DAO's token holders (Board Members), and actions are executed via proposals enforced by on-chain voting.

3.2 Access-Control

In addition to the access control protocol employed by the `TimeLock` contract, the system implements a Role-Based Access Control (RBAC) [18] protocol focused on DAO members. The DAO has three main member roles: (i) BOARD_MEMBER, (ii) SUPPLIER_MEMBER and (iii) SENIOR_SUPPLIER_MEMBER. The board and supplier member's roles are created and managed by the `AdministrativeAccessControl` contract, while the upgrade to a senior supplier is managed by the same contract that deals with the supplier's service proposal logic, the `SupplierProcess` contract.

The DAO is deployed with a configurable number (v) of pre-decided board members that are the sole voting entities of this community. This power is exclusive to the board members (role BOARD_MEMBER) because the objective of the DAO is to provide an unbiased service between the Principal and the Agent of a contracting transaction. In this case, the Principal is represented by the jury members of the public institution, who have the right to have complete control over the contracting decision. Board members have the exclusive power to:

- vote on proposals;
- start new procurement contests;
- add/revoke suppliers;
- add/revoke board members;
- mint Governance Token.

Suppliers are the Agent in the contracting transaction and aim to participate in procurement contests. After a successful proposal to add a new supplier, the provided address is assigned the supplier role. Supplier members have the exclusive capability to submit contest applications to a respective procurement contest. This logic can be found in the `SupplierProcess` contract. A supplier earns one reputation point per contest won. After a configurable number of victories (w), the DAO automatically upgrades the supplier to senior supplier by granting it the role of SENIOR_SUPPLIER_MEMBER. Senior suppliers have all the capabilities of a regular supplier with the upgrade of being able to submit feedback proposals that, if approved, are applied by the board members, giving a voice to applicants that provide an excellent service to the public institution. This operation is managed by the `SeniorSupplierProcess` contract presented in Listing 1. Furthermore, to decentralise the board members' power over the suppliers, only a supplier or senior supplier can propose revoking a senior supplier's role.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";

contract SeniorSupplierProcess is Ownable {
  string private feedback;

  // Emitted when the stored value changes
  event NewFeedback(string newFeedback);

  // Stores a new Feedback in the contract
  function newFeedback(string memory _newFeedback) public
      onlyOwner {
    feedback = _newFeedback;
    emit NewFeedback(_newFeedback);
  }

  // Reads the last stored Feedback
  function retrieve() public view returns (string memory)
      {
    return feedback;
  }
}
```

Listing 1: SeniorSupplierProcess.sol contract code

An alternative would be to limit the proposals in which board members could vote and provide limited voting power to senior suppliers, allowing them to vote on exclusive senior supplier proposals. However, this approach would result in the creation of a second board of senior suppliers. This new board would require an

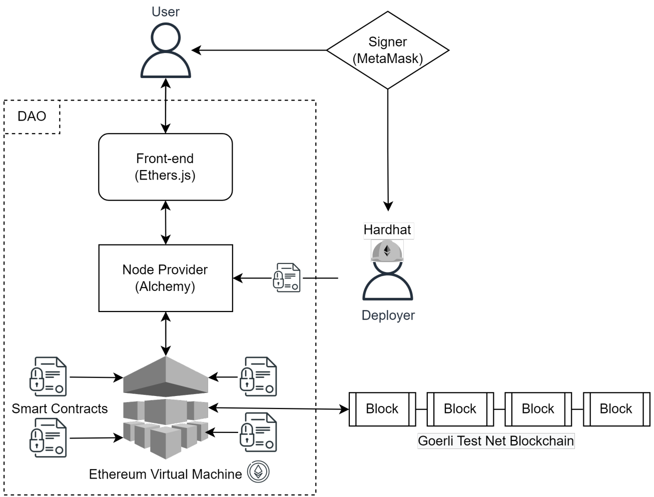

Figure 4: Public Procurement DAO Framework Architecture

additional governance protocol resulting in two DAOs interacting with each other rather than two entities interacting with one DAO.

Although a valid alternative, it presents problems concerning the distribution of powers, where a group of senior suppliers could monopolise the contracting system against standard suppliers. It would also create two rival communities instead of one more vast community where members are combined to achieve the most profitable outcome.

3.3 Implementation

This system follows a standard Web 3.0 DApp architecture with three central units:

- node provider;
- transaction signer;
- smart contracts.

A blockchain is a distributed system with nodes that keep the state of the program; when a user needs to communicate with a smart contract, its client contacts one of these nodes. For experimenting with a blockchain, the researcher has two alternatives. The first is to set up his own node and run it himself. A complete solution; however, launching a blockchain infrastructure from scratch is complex and becomes more challenging when future scalability is necessary. Therefore, this project implements the second option, using a node provider. A node provider grants read and write access to the blockchain by providing a web JSON-RPC-based API [2]. There are several third-party node providers; in this project, we chose *Alchemy* due to its compatibility with Ethereum blockchains, its high reliability compared to competitors, and its proven track record of data accuracy.[1] Using a blockchain node provider, this project reduces maintenance costs and improves reliability.

In a blockchain, each request is a transaction that must be signed with the client's private key. Therefore, a transaction signer such as *MetaMask* is required.[2] In this system, we used MetaMask to create the necessary wallet accounts to deploy and test the DAO. MetaMask stores the client's address and private key in the browser and, connected to the blockchain, signs each transaction request made by a client to the DAO.

Although clients use their wallets to interact with the DAO, they do not make that interaction directly with the smart contracts. This solution implements a front-end user interface with bash scripts and a compact library called *Ethers.js*, which enables DAO deployment and interaction with an Ethereum blockchain. The described architecture is represented in Figure 4.

The project was also developed using the *Hardhat* framework.[3] *Hardhat* is a development environment for Ethereum software. It consists of different components to edit, compile, debug, and deploy smart contracts and DApps [12].

4 EVALUATION

Ethereum is a permissionless blockchain, which means that anyone can join and use the infrastructure [19]. To avoid misuse, Ethereum charges Ether for the execution of smart contracts, making it inconvenient to use when evaluating applications. Because of that, Ethereum test networks or testnets emulate Ethereum's behaviour and do not require actual payment; instead, transactions are paid through false Ether obtained for free through a faucet.

Goerli is a Proof-of-Stake Ethereum testnet [11]. This testnet was chosen to test the system because the Ethereum mainnet recently changed its consensus to Proof-of-Stake and deprecated most other testnets. *Hardhat* also recommends the use of *Goerli* due to its stability.

This evaluation assesses four performance metrics:

- ***Contract Size***: The size of the Solidity contracts' EVM bytecode in kilobytes, obtained through a *Hardhat* plugin named *Contract Sizer*.
- ***Gas Price***: The cost per unit of gas specified for a transaction in *Ether*. The higher the gas price, the higher the chance of being included in a block. Obtained in *Etherscan*.
- ***Transaction Fee***: The product of the gas used and the gas price paid to the block producer in *Ether* to process the transaction. Obtained through *Etherscan*.
- ***Latency***: The average time in seconds to complete an operation, from the client's request to the system's response. Obtained through *Hardhat*.

4.1 System Deployment

The first step in the launch of a DApp is the deployment of its smart contracts. The deployment of a smart contract consists of a transaction containing the compiled code of the smart contract without specifying any recipient [10].

The completion time of an Ethereum transaction depends on how much gas is paid (i.e., transaction fees) and how congested the Ethereum network is. This system was deployed on 22^{nd} November 2022 in the Goerli testnet. Therefore, we must take into account the standard network values on that day, such as the average Ethereum transaction fee of $0.000464 Eth$ and the standard gas price of 1.935×10^{-9} ETH [24].

[1] https://www.alchemy.com/
[2] https://metamask.io/
[3] https://hardhat.org/

Contract Name	Size (KiB)	Gas Price (Ether)	Transaction Fee (Ether)
AdministrativeAccessControl	5.897	2.0024×10^{-8}	0.03574
AdministrativeRecruitmentProcess	1.312	2.3018×10^{-8}	0.008481
GovernanceProtocol	18.943	2.0117×10^{-8}	0.09037
GovernanceToken	8.842	2.2581×10^{-8}	0.04706
SeniorSupplierProcess	1.312	2.3525×10^{-8}	0.008667
SupplierProcess	6.164	2.2469×10^{-8}	0.03239
TimeLock	7.438	2.5007×10^{-8}	0.04722
Total	49.908	–	0.2699

Table 1: Deployment Evaluation Results

Table 1 shows the sizes of the bytecode of all contracts in kilobytes and their respective deployment costs in *Ether*. When analysing these values, it is possible to draw several important conclusions.

First, by comparing the `GovernanceProtocol` contract (18.943KB) with `SeniorSupplierProcess` (1.312KB), we can see a correlation between the complexity and size of each contract, where the larger contract holds most of the system logic, and the smaller one carries only two simple functions.

Second, all contracts respect the size limit of 24KB on bytecode size of smart contracts in Ethereum, introduced on the EIP 170, with the help of the Ethereum Optimizer.

It is also possible to conclude that the larger the contract, the higher the cost of deployment. This can be observed by comparing the deployment of `GovernanceProtocol.sol`, the largest contract of the system (18.842 KiB), which cost approximately 0.09 ETH, against the smallest, `SeniorSupplierProcess.sol` (1.312 KiB), which only cost approximately 0.008 ETH.

Finally, we can analyse the total cost of deploying the entire system, approximately 0.27 ETH (301,78€ on 30/12/2022), and accept that it is an acceptable amount for a one-time deployment. Of course, such values must take into account the volatility of Ethereum.

4.2 System Performance

This section of the evaluation process examines each system operation and analyses their respective transaction fees and latency values.

It is necessary to remember that the time it takes for an Ethereum transaction to complete depends on how much gas is paid and how congested the Ethereum network is. Paying the standard gas price (2.689×10^{-8} ETH on 17/01/2023 [24]) takes, on average, 15 seconds to 5 minutes to process a transaction. The executions used in this evaluation were made on 17^{th} and 18^{th} January 2023 and are therefore compared with the average Ethereum metrics of those days.

With this in mind, it is possible to reach some meaningful conclusions about the performance evaluation of this system.

Table 2 shows the average transaction gas prices and execution fees in *Ether* and the average latency of each operation in seconds.

The first reflection is that the average gas price paid per transaction was approximately 3.3746×10^{-8} ETH, higher but very close to the average on both testing days, 3.0225×10^{-8} ETH. Therefore, we can accurately compare the DAO's average transaction fees and latencies. The average transaction fee of the DAO's operations is 0.004060 ETH, substantially higher than the Ethereum average of 0.000395 ETH.

Regarding the latency values, Table 2 shows that regardless of similar gas prices, all operations present a higher latency value than the Ethereum average. Analysing the data in Table 2, we can confirm a difference of 5.71s with a system average of 17.76s and an Ethereum average of 12.05s.

5 CONCLUSION

With the analysis of a practice used by many democratic governments to select private institutions to provide services, this study presents an approach that improves such a practice based on the benefits of blockchain technology.

The core problem analysed focuses on a business interaction between a Principal and an Agent and the implementation of trusted third parties in such negotiations. This study found solutions to those problems in the characteristics of Decentralised Autonomous Organizations.

Based on the permissionless blockchain, Ethereum, the presented system contains all the necessary use cases of a standard public procurement contest. The DAO is deployed with a board of members, representing the public institution, that holds the voting power. These board members propose the admission of new suppliers. Suppliers can propose their services to specific procurement contests launched by board members of the public institution. Board members can then approve or reject, in a majority, supplier proposals.

The presented system was developed in Solidity and JavaScript, deployed in the Goerli testnet, and follows the same workflow of standard public procurement contests. By introducing decentralisation, security, and transparency attributes, this study demonstrates a solid solution that deprecates trusted third parties in public procurement contests and exhibits a new application of DAOs in today's society.

Although this is a solid first proposal, many considerations must be made if this solution is to be implemented in a real-world scenario. Potential challenges and obstacles must be identified and addressed to mitigate negative consequences or unintended outcomes. Here we present some of those considerations.

The first involves the senior applicants and their exclusive power to submit feedback proposals regarding the contests and overall functionality of the DAO. Usually, critiques of a public procurement contest can be suggested by anyone rather than only from past successful applicants. In fact, successful applicants have a more significant probability of remaining silent, as they are content and may

Operation	Gas Price (ether)	Transaction Fee (ether)	Latency (s)
Administrative Proposal	5.3556×10^{-8}	0.006308	20.02
Supplier Proposal	6.8302×10^{-8}	0.008709	16.22
Mint Token	3.1515×10^{-8}	0.003099	15.48
Vote	1.7244×10^{-8}	0.001987	20.66
Queue	1.5472×10^{-8}	0.002111	16.45
Execute	1.6384×10^{-8}	0.002148	17.71

Table 2: Performance evaluation average results

want to ensure that certain conditions in a procurement contract that favoured them in the past remain. Usually, most suggestions come from unsuccessful applicants that tend to have valid concerns about fairness.

Another consideration is that the presented system only takes one round to select a winner among all applicants. In a real-world scenario, a proposal/application is unlikely to match the procurement specification precisely. In such cases, major and minor deviations need to be considered. Usually, each application is given a score based on such significant deviations. Each jury member cannot do this calculation individually as they need each other's expertise. Such calculations should also be public information rather than just individual votes.

The last and most important consideration is that in real-world public procurement, the application's price and some aspects of the proposal (e.g., technical specification of the proposed product) should only be made public once the winner is determined. In this solution, the entire proposal will be public for transparency reasons. It is recommended to consider well-known information-hiding techniques proposed for blockchains in such cases.

In conclusion, the proposed system seeks to address the challenges that current public procurement systems face by introducing a new DAO model and incorporating the benefits of blockchain technology. The objectives of the system of reducing the involvement of third-party contractors, increasing traceability and transparency, and improving the efficiency of the negotiation process will ultimately contribute to a more efficient and effective public procurement process.

This work is expected to be a baseline upon which other educational institutions and public organisations can build.

ACKNOWLEDGMENTS

This work was supported by the European Commission under project BIG ERA Chair (grant agreement 952226) and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID).

REFERENCES

[1] Ledger Academy. 2021. Your DAO guide: The Most Important DAO Categories Defining the Space. https://www.ledger.com/academy/your-dao-guide Last accessed 30 November 2022.
[2] Alchemy. 2022. What is Ethers.js? https://docs.alchemy.com/docs/what-is-ethers-js Last accessed 15 December 2022.
[3] Andreas M. Antonopoulos and Gavin Wood. 2019. *Mastering Ethereum* (1 ed.). O'Reilly Media Inc, CA.
[4] Vitalik Buterin. 2014. DAOs, DACs, DAs and More: An Incomplete Terminology Guide. https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide Last accessed 28 November 2022.
[5] Wei Cai, Zehua Wang, Jason B.Ernst, Zhen Hong, Chen Feng, and Victor C. M. Leung. 2018. Decentralized Applications: The Blockchain-Empowered Software System. *IEEE Access* 6 (2018), 53019–53033.
[6] Jakša Cvitanić and Jianfeng Zhang. 2013. *Principal–Agent Problem*. Springer.
[7] Youssef El Faqir, Javier Arroyo, and Samer Hassan. 2020. An overview of decentralized autonomous organizations on the blockchain. In *Proceedings of the 16th International Symposium on Open Collaboration*. 1–8.
[8] Ethereum. 2021. Introduction to Smart Contracts. https://docs.soliditylang.org/en/v0.8.17/introduction-to-smart-contracts.html Last accessed 28 November 2022.
[9] Ethereum. 2022. ERC-20 Token Standard. https://ethereum.org/en/developers/docs/standards/tokens/erc-20/ Last accessed 05 January 2023.
[10] Ethereum.org. 2022. Deploying Smart Contracts. https://ethereum.org/en/developers/docs/smart-contracts/deploying/. Last accessed 06 January 2023.
[11] Goerli. 2022. Goerli Testnet. *Goerli Website* (2022). https://goerli.net/ Last accessed 16 December 2022.
[12] Hardhat. 2022. Hardhat Runtime Environment (HRE). https://hardhat.org/hardhat-runner/docs/advanced/hardhat-runtime-environment Last accessed 15 December 2022.
[13] Evan Jones. 2022. Review of Uniswap Exchange. https://www.cryptovantage.com/best-crypto-exchanges/uniswap/ Last accessed 30 November 2022.
[14] Ministério das Obras Públicas, Transportes e Comunicações. 2008. Decreto-Lei n.º 18/2008. https://dre.pt/dre/detalhe/decreto-lei/18-2008-248178
[15] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. *bitcoin.org* (2008), 3–4. https://bitcoin.org/bitcoin.pdf
[16] OpenZeppelin. 2022. Governance. *OpenZeppelin Docs* (2022). https://docs.openzeppelin.com/contracts/4.x/api/governance Last accessed 07 December 2022.
[17] Morgen E Peck. 2017. Blockchains: How they work and why they'll change the world. *IEEE Spectrum* 54, 10 (2017), 26–35.
[18] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. 1996. Role-based access control models. *Computer* 29, 2 (1996), 38–47.
[19] Diogo Serranito, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2020. Blockchain ecosystem for verifiable qualifications. In *2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 192–199.
[20] Alexandra Sims. 2019. Blockchain and Decentralised Autonomous Organisations (DAOs): The Evolution of Companies? *University of Auckland Business School* (November 2019). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3524674 Last accessed 30 November 2022.
[21] Madhusudan Singh and Shiho Kim. 2019. Blockchain technology for decentralized autonomous organizations. In *Advances in Computers*. Vol. 115. Elsevier, 115–140.
[22] Philip Treleaven, Richard Gendal Brown, and Danny Yang. 2017. Blockchain Technology in Finance. *Computer* 50, 9 (2017), 14–17. https://doi.org/10.1109/MC.2017.3571047
[23] Shuai Wang, Wenwen Ding, Juanjuan Li, Yong Yuan, Liwei Ouyang, and Fei-Yue Wang. 2019. Decentralized autonomous organizations: Concept, model, and applications. *IEEE Transactions on Computational Social Systems* 6, 5 (2019), 870–878.
[24] YCharts. 2023. Ethereum Average Gas Price. https://ycharts.com/indicators/ethereum_average_gas_price Last accessed 02 January 2023.