



<b>Title</b>	Towards a Blockchain Implementation of a Governance & Revenue Dispersal Mechanism for Investments in Battery Energy Storage Systems
<b>Authors(s)</b>	Bird, Bryan, Bokkissam, Hanumantha Rao, Savelli, Iacopo, Morstyn, Thomas, Cuffe, Paul
<b>Publication date</b>	2023-06-29
<b>Publication information</b>	Bird, Bryan, Hanumantha Rao Bokkissam, Iacopo Savelli, Thomas Morstyn, and Paul Cuffe. "Towards a Blockchain Implementation of a Governance & Revenue Dispersal Mechanism for Investments in Battery Energy Storage Systems." IEEE, June 29, 2023. <a href="https://doi.org/10.1109/PowerTech55446.2023.10202851">https://doi.org/10.1109/PowerTech55446.2023.10202851</a> .
<b>Conference details</b>	The 2023 IEEE Belgrade PowerTech, Belgrade, Serbia, 25-29 June 2023
<b>Publisher</b>	IEEE
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/25737">http://hdl.handle.net/10197/25737</a>
<b>Publisher's statement</b>	© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
<b>Publisher's version (DOI)</b>	10.1109/PowerTech55446.2023.10202851

Downloaded 2026-05-02 00:29:53

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

# Towards a Blockchain Implementation of a Governance & Revenue Dispersal Mechanism for Investments in Battery Energy Storage Systems

Bryan Bird

*School of Electrical and Electronic Engineering  
University College Dublin  
Dublin, Ireland  
birdbjm@gmail.com*

Hanumantha Rao Bokkissam

*School of Electrical and Electronic Engineering  
University College Dublin  
Dublin, Ireland  
bokkissam.hanumantharao@ucd.ie*

Iacopo Savelli

*School of Engineering Science  
University of Edinburgh  
Edinburgh, UK  
isavelli@ed.ac.uk*

Thomas Morstyn

*School of Engineering Science  
University of Edinburgh  
Edinburgh, UK  
thomas.morstyn@ed.ac.uk*

Paul Cuffe

*School of Electrical and Electronic Engineering  
University College Dublin  
Dublin, Ireland  
paul.cuffe@ucd.ie*

**Abstract**—The rapid onboarding of renewable generation has led to many difficulties for both generators and the grid due to volatile output and market conditions. Battery energy storage systems have begun to alleviate these issues. However, such systems face many financial challenges in their deployment and operation. This paper demonstrates the use of advanced distributed ledger technology with smart contracts to design a decentralised autonomous organisation (DAO) to manage the investments in these systems and optimise their operations. In this paper, a series of smart contracts have been designed, deployed, and tested to demonstrate the effectiveness of such a DAO framework for battery management. The interconnected smart contracts can manage the financing, governing, and revenue-sharing mechanisms, allowing investors to coordinate investment and vote on the battery energy storage system's operation (charge/float/discharge) for a given period. This interconnected smart contract design implementation demonstrates a novel and effective strategy for managing battery energy storage systems, decreasing barriers to entry for investors, thus, allowing a greater adoption of such technology.

**Keywords**—Battery energy storage system, Blockchain, Decentralised governance, Distributed Autonomous Organisation, and Smart contracts.

## I. INTRODUCTION

The onboarding of renewable energy to global energy grids has increased exponentially in recent years. While this is a necessary transition to tackle climate concerns, significant technical challenges are faced with this shift in generation. Foremost of these problems come from the intermittent nature of these energy sources. This problem creates a constraint on energy production and onboarding grid conditions limiting the penetration of these generations into the market [1]. Significant losses of potential renewable energy and increased grid operation expenses are experienced due to curtailment costs used to coordinate this generation with demand [2]. A solution to this issue that has been growing in popularity is a battery energy storage system (BESS).

The BESS has begun to be utilised more frequently at a grid level due to improvements and reduced battery technology

costs, finding particular traction in micro-grids for economic management of peak demand [3]. While BESS has seen wider adoption with grid operations, the initial investment is still high at larger scales, and revenue flows can fluctuate depending on the grid conditions. For this reason, BESS deployments will seek to optimise battery operations by choosing the best charging (discharging) strategies when spot market prices are low (high) [4]. These charging/discharging methods can result in vastly different financial returns for operators. As a result of grid conditions and economic incentives, investors of BESS seek to collaborate on larger battery installations to maximise returns [5].

This grouping of investors leads to a coordination problem of managing a potentially distributed group of investors, a crucial issue from an investor's standpoint and the grid operator. Therefore novel mechanisms are required to address these investments and physical systems. There are different BESS configurations, such as shared residential battery energy storage, shared local battery energy storage, and shared virtual energy storage. For instance, shared residential battery energy storage of up to perhaps 20kWh is shared among the community members of a specific location, for example, Gridflex Heeten [6]. Shared local battery energy storage shares hundreds of kWh with community ownership and governance, for instance, Feldheim energy community [7]. Shared virtual battery energy storage is decentralised and installed at different locations with independent ownership and governance, for example, SonnenCommunity [8].

An equally fast-growing development that could provide a suitable framework for managing BESS investments is distributed ledger technology (DLT) [9]. These distributed systems include blockchain technologies, which offer decentralised operations of small-scale systems secured by cryptographic practices ideally suited for peer-to-peer mechanisms [10]. A primary advantage of a blockchain is its ability to operate secure and without the intervention of intermediaries. Thus tokenisation has proven incredibly impactful in finance and supply chain and has begun to see practical adoption in

these industries [11]. While many attempts have been made to tokenise energy units directly, the difficulty of accurately tracking and securely reporting consumption and generation for use with blockchain technology [12], coupled with regulatory uncertainty, has limited its application beyond micro-grids [13]. An equally promising area for blockchain application is governance and system coordination. Through smart contracts, code deployed and executed on a blockchain, decentralised autonomous organisations (DAOs) have been deployed to many organisational use cases [14].

A DAO allows users to make proposals, store information, and manage voting through different voting mechanisms to either agree on organisational decisions or manage to underline assets and investments. [15] [16]. Tokenization enables votes within a DAO to be cast and managed securely and transparently. This process has found success within various industries. For example, projects such as Uniswap, which operates the world’s largest Decentralised Exchange [17] or LexDAO, legal professionals seeking to use smart contracts to automate legal services [18]. These various entities have utilised a DAO model to help further coordination between participants beyond a traditional incorporation model and at much lower costs.

This paper seeks to deploy this already-proven concept of DAO governance to managing a BESS and its charging/discharging mechanism. This governance structure would allow for a lightweight and decentralised approach to battery operation and revenue management, tackling many of their current difficulties. The DAO framework’s operational rules are underpinned by a series of inter-connected smart contracts based on predefined rules that self-execute. The BESS can secure revenue streams by directly participating in an energy arbitrage mechanism governed by a DAO. The revenues generated from the battery by participating in the energy arbitrage strategy, and directly possessed on-chain by the DAO itself, are ultimately disbursed to the decentralized shareholders based on the pro-rata share of their investments.

In [19], the authors have presented a case study simulation of this radical conception of asset ownership and control, whereby token holders vote based on the real-time price of electricity, their particular electricity price forecast, and the current state of charge of the battery. The authors have investigated the impact such a mechanism could have on revenue streams from battery management. In [19], the authors did not develop the software implementation of such smart contracts, which motivates the design and development of software implementation outlined and tested below will be structured and developed based on their findings and recommendations. The software implementation of such a DAO framework for BESS governance with interconnected smart contracts is the significant contribution of this paper.

## II. ARCHITECTURE OF PROPOSED SYSTEM

The DAO framework has been conceptualised as a system of interconnected smart contracts that govern the BESS operation and onward revenue sharing. Validating the feasibility of such a framework requires developing a series of prototype smart contracts for managing the proposals for battery operation, voting based on token holdings, and revenue management of any associated funds. Various tools and frameworks have been utilised to deploy these contracts to a public blockchain for testing and validation. The proposed architecture of the

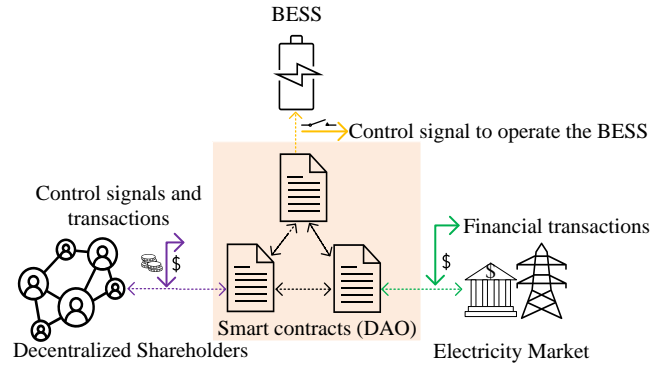


Fig. 1. Architecture of proposed DAO framework.

### Algorithm 1 Step-by-step procedure for smart contract operation.

- 1: BESS Investor purchases tokens from the finance smart contract function call for voting rights.
- 2: A proposal for a charging/discharging/floating method is created on the governance contract by the investor.
- 3: Investors vote for the said proposal, and it is executed.
- 4: After funds are released by revenue contract after the withdrawal proposal is passed and the payout function is called by governance smart contract.

prototype DAO framework for financing, governing, and disbursing revenues of a battery energy storage system is shown in Fig. 1. In this framework, three interconnected smart contracts (*financing* smart contract, *governing* smart contract, and *revenue-sharing* smart contract) autonomously interact with the token holders, dispatches the physical battery itself, and financially transacts with the wholesale electricity market based on predefined rules and conditions. The financing smart contract interacts with investors to finance the BESS. The physical control signal to operate the battery is mediated through the governing smart contract controlled by shareholders’ votes for operation (charge/discharge/float). The revenue flows between financial electricity markets and token holders are mediated through a revenue-sharing smart contract.

## III. SMART CONTRACT DEVELOPMENT METHODOLOGY

The prototype smart contracts have been designed and developed using the Solidity programming language and deployed to an Ethereum test network [20], [21]. The code has been executed based on set functions and stored variables. The proposals, casting votes, and project information are held and managed through smart contract function calls broadcast to the blockchain network. The open-source code is available to view through any block explorer, such as *Etherscan*. The *hardhat* framework has been used to write and deploy these contracts, which provides a set of deployment scripts and project management tools for Solidity and blockchain projects[22]. Additionally, The REMIX IDE is used for granular testing and validation. It is a browser-based environment for managing contract calls and testing EVM coding for more granular interactions and optimisation [23].

The deployed contracts are built upon the *OpenZeppelin* library [24], a series of open-source smart contracts. These

contracts are beneficial for implementing governance standards, such as proposals and voting mechanisms. *OpenZeppelins* series of governance-based contracts are utilised to manage DAO operations such as proposals. These open-source functions offer tested and verified voting and proposal management implementations that can be extended for specific use cases. For the current implementation, all contracts are deployed to the Goerli Test-nets, a live test environment used to test implementations of smart contract code before deployment to the Ethereum Mainnet.

Algorithm 1 demonstrates the series of steps for the minimal operation of a BESS DAO and the capability of the current system. The three deployed contracts can oversee the management of funds and operation of the battery system with minimal effort from investors. The four steps outlined above and associated transactions allow for prototype functionality of the proposed BESS DAO. These steps can be repeated at set intervals to operate the charging/discharging of a BESS.

## IV. RESULTS & DISCUSSIONS

### A. Finance Smart Contract

The finance smart contract consists of a basic implementation of *open-zeppelins ERC-20 token* standard. This smart contract controls the supply and balances of these tokens for use within all governance carried out within the BESS DAO. The DAO shareholder can purchase tokens directly from this contract using the “*buyTokens*” function as shown in Listing 1.

Listing 1. *BUYTOKENS*

```
//BuyTokens function allows the caller to buy a set
//amount of tokens by sending a transaction with
//the amount and corresponding Ether.
function buyTokens(uint256 _amount) public
payable{
    //calculates the cost of tokens
    uint256 cost = tokenPrice*_amount;
    //checks the user has sent enough Ether and
    //that the contract has received enough
    //Ether.
    require(msg.value == cost, "Insufficient
    Payment");
    require(address(this).balance >= cost, "
    Insufficient Funds");
    //Calls to the revenue contract to update
    //the balance
    revenue.setDeposit(msg.sender,msg.value);
    //transfers funds to the revenue contract
    //based on amount purchased
    (bool success, ) = address(revenue).call{
    value: _amount}("");
    require(success, "Failed to send ether");
    //mints new tokens to the purchasers
    //wallet address and validates the
    //transfer
    _mint(msg.sender, _amount);
    require(balanceOf(msg.sender) >= _amount, "
    Failed to buy tokens");
}
```

This function allocates the ERC-20 tokens purchase voting rights and membership of this system to the buyer’s address. All funds received from this contract are then transferred to our revenue contract for storage and management using an interface call.

### B. Governance Smart Contract

The governing smart contract is the primary smart contract for the operation of the DAO. It allows users to propose vote and implement proposals using ERC-20 tokens purchased from the finance contract as votes. Equally, it mediates the release of funds from the revenue smart contract based on proposals and votes. Standard DAO functionality is imported, including proposal structures, quorum percentages, and vote tracking. For this reason, much of the code consists of override functions that implement the base functionality of imported functions and override them with new values for use within our implementation [25].

Listing 2. *PROPOSE*

```
//Propose takes specific arguments for setting up a
//proposal. This override function extends the
//functionality of OpenZeppelins implementation to
//our contract.
function propose(
    address[] memory targets,
    uint256[] memory values,
    bytes[] memory calldatas,
    string memory description
) public override(Governor) returns (uint256) {
    // here, variables are passed to the base
    // contract implementation, and the proposal ID
    // is returned once this data is stored.
    return super.propose(targets, values, calldatas,
    description);
}
```

Proposals are cast by shareholders using the propose function in Listing 2. These proposals consist of the charging method used within the battery system for the following hour. Once cast, they are to be open for voting for a period counted in seconds. Other shareholders can then use the vote function to allocate their votes to their preferred charging method for that period or any other proposal created for the operation of the DAO. Once the quorum percentage is reached for a proposal or the time runs out, the most popular proposal will be passed and can be executed. While the execution is implemented in this deployment, its demonstration is beyond the scope of this paper due to the need for support infrastructure and physical battery assets. Currently, voting is carried out in a yes-no manner, with tokens being used to validate membership. However, the contract supports more robust measures, such as additional voting options, weighted averages of votes, or the ability to adjust the quorum percentage. In this way, the governance model utilised can be adjusted to suit the needs of the DAO with little effort or redeployment.

### C. Revenue-sharing Smart Contract

The revenue-sharing smart contract operates as an escrow account for storing all funds managed by the DAO. The purchase of tokens, tokenised revenue from the sale of battery-stored energy, and this contract manages any other forms of revenue. Updated DAO membership is received from the finance contract while the governance mediates the release of any funds. The *makePayment* function allows for the automated payout of funds to the users, as shown in Listing 3. This function is managed by the Governance smart contract and can only be called by this address once the DAO has successfully passed a proposal. The function loops over the membership, transferring a share to each participant. This is currently distributed evenly among

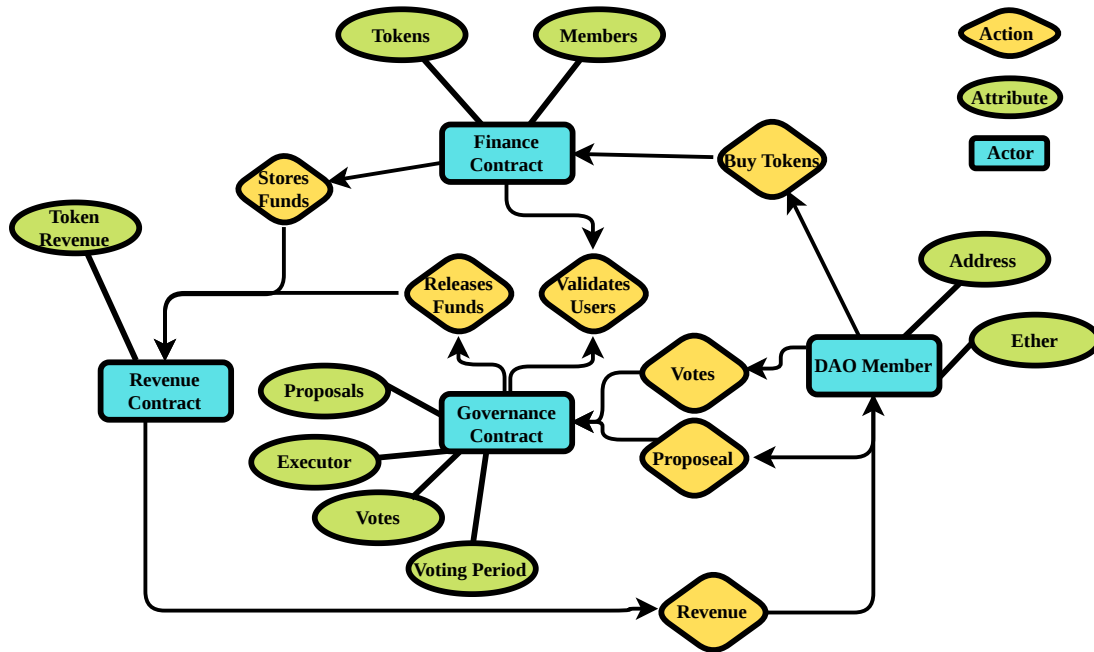


Fig. 2. Entity relationship diagram for a BESS autonomous governance.

shareholders but can be based on a percentage of tokens held by each individual or other such novel arrangements.

Listing 3. MAKEPAYMENT

```

// MakePayment function is called by the governance
// contract once a proposal is passed and executed
// for withdrawing funds

function makePayment() external payable {
    //amount is calculated by dividing the
    //balance of the contract by the number of
    //shareholders
    uint256 _amount = (address(this).balance)/(
        Members.length);
    //Validates that the message sender is the
    //governance contract. Only calls executed
    //by the DAO after voting is able to
    //withdraw funds.
    require(msg.sender == address(Governance), "
        You are not the DAO");
    //Loops over all members addresses and
    //transfers their share of the funds.
    for(uint i=0; i<Members.length; i++){
        (bool success, ) = payable(address(
            Members[i])).call{value: _amount}("");
        require(success, "Failed to send ether")
    }
}

```

1) *Simulated DAO Operation:* This DAO governance system has been deployed and tested on the *Goerli test net* with the contracts being available to view through Etherscan. A simulated demonstration of simple token purchase, proposal, voting, and the subsequent payout was carried out to test the implementation of this design. Below is an outline of this process, with each associated transaction. This simulation assumes that a DAO has been put in place to control a BESS, and shareholders are voting for their preferred charging/discharging mechanisms based on market conditions every hour.

For testing purposes, proposals were carried out in a shorter time frame. This simulation consisted of the minimum number of interactions required to test all functionality of the contracts in a real-world setting. Costs associated with the deployment and operation of these contracts have been estimated. These costs are important, as all operations require transaction fees to be paid for execution. Transaction costs associated with the above deployments of the three contracts amounted to 0.007172 Ether or approximately \$8.83 at the time of writing. The current example shows only the interaction and sequence of events that a DAO member may experience when interacting with this system. More complex voting schemes are not shown. Table 1 shows the transaction details for each step with associated Etherscan links. Accompanying diagrams are all derived from the data flows of the test system.

As demonstrated in Table I standard operation of the DAO can be broken down into two phases Set up and execution. The setup phase constitutes the required steps to begin voting and actions being executed by the DAO. The operation phase consisted of voting by shareholders and execution of these proposals, including the release of funds. The steps outlined in the operation phase will be repeated at regular intervals, such as once hourly, for management of the BESS and its charging/discharging method.

The entity relationship diagram shown in Fig. 2. highlights the interactions between these three contracts and associated variables. As seen in both figures, the governance contract is central to the operation of this contract. This contract manages most data and is the primary interaction source for any DAO member. The sequence interactions for this system are visualized in Fig. 3. Minimal DAO operations are displayed by showing a DAO member's interaction with these interconnected smart contracts. Each interaction from the investor represents a single transaction and its child processes within the system.

TABLE I.  
EXAMPLE TRANSACTION SEQUENCE FOR SETUP AND ONGOING GOVERNANCE OF BESS DAO

Phase	Step	Transaction Purpose	Record	Block ID	Cost in Ether
→ Set Up	1	Purchase of tokens by DAO member for voting rights	[26]	8226409	0.000478
	2	Creation of proposal for charging/discharging method	[27]	8226454	0.000234
	3	Voting on proposal for charging/discharging method	[28]	8226466	0.000158
○ Operation	4	Proposal for the release of revenue to DAO shareholders	[29]	8226485	0.000235
	5	Transaction for the release of revenue to DAO shareholders	[30]	8226520	0.000107

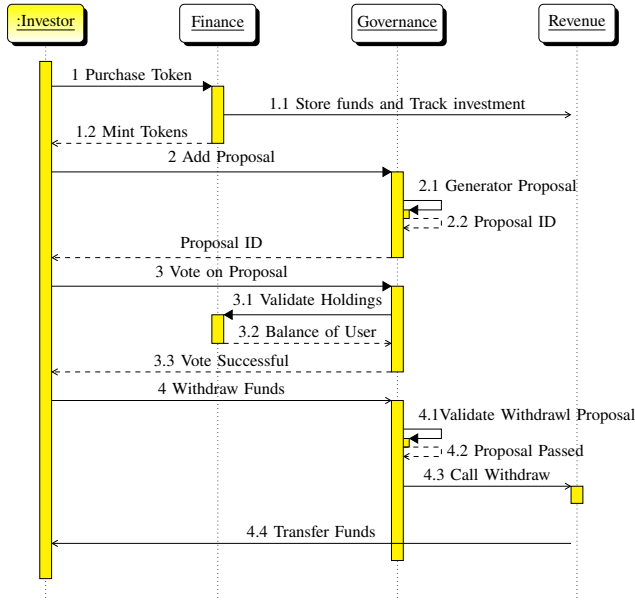


Fig. 3. Sequence diagram for DAO operation.

## V. CONCLUSIONS

This paper has presented the design, development, and testing of a prototype distributed autonomous organisation (DAO) framework with interconnected smart contracts for the financing, governing, and revenue dispersal of a battery energy storage system. The outline software implementation of such a DAO is the significant contribution of this paper. The previous approaches in the literature have focused mainly on applying blockchain technology to realising tokenised quanta of electrical energy. However, by concentrating on tokenisation of other aspects of energy production, such as governance over real-world assets and coordination of investors, many problems with enforcement can be avoided, potentially leading to optimised returns and better grid operation. As with all blockchain applications, access to real-world data is limited due to the static nature of smart contracts and their inability to trustlessly access external data during code execution. As such, direct management of the battery system would need to be handled by secondary system reading and writing to and from the blockchain for direct control of the battery and its charging mechanism. Using a DAO to manage a battery energy storage system may allow for more effective coordination of investors; such a system might decrease barriers to adopting this technology and enables a greater onboarding of renewable energy.

## ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 20/EPSC/3700 and EPSRC under Grant number EP/S031901/1.

## REFERENCES

- [1] P. Denholm and M. Hand, "Grid flexibility and storage required to achieve very high penetration of variable renewable electricity," *Energy Policy*, vol. 39, no. 3, pp. 1817–1830, 2011.
- [2] M. Joos and I. Staffell, "Short-term integration costs of variable renewable energy: Wind curtailment and balancing in Britain and Germany," *Renewable and Sustainable Energy Reviews*, vol. 86, pp. 45–65, 2018.
- [3] A. Muvvala, S. Kotra, and M. K. Mishra, "Economic power management with BESS," in *2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)*, 2019, pp. 024–029. doi: 10.1109/ICSETS.2019.8745177.
- [4] X. Li, D. Hui, and X. Lai, "Battery energy storage station (BESS)-based smoothing control of photovoltaic (PV) and wind power generation fluctuations," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 2, pp. 464–473, 2013. doi: 10.1109/TSTE.2013.2247428.
- [5] P. Neetzow, R. Mendelevitch, and S. Siddiqui, "Modeling coordination between renewables and grid: Policies to mitigate distribution grid constraints using residential PV-battery systems," *Energy Policy*, vol. 132, pp. 1017–1033, 2019.
- [6] Gridflex, *Gridflex 2022*. [Online]. Available: <https://gridflex.nl/>.
- [7] NEFF, *New energies forum feldheim 2022*. [Online]. Available: <https://nef-feldheim.info/?lang=en>.
- [8] SonnenCommunity, *Sonnencommunity. sonnen 2022*. [Online]. Available: <https://sonnen.de/sonnencommunity/>.
- [9] N. El Ioini and C. Pahl, "A review of distributed ledger technologies," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, Springer, 2018, pp. 277–288.
- [10] M. Anderson, "Exploring Decentralization: Blockchain Technology and Complex Coordination," *Journal of Design and Science*, Feb. 2019, <https://jods.mitpress.mit.edu/pub/7vxemtm3>.
- [11] T. K. Agrawal, J. Angelis, W. A. Khilji, R. Kalaiarasan, and M. Wiktorsson, "Demonstration of a blockchain-based framework using smart contracts for supply chain collaboration," *International Journal of Production Research*, pp. 1–20, 2022.
- [12] A. de Villiers and P. Cuffe, "Dissolving Metaphors in Peer-to-Peer Energy Trading: Towards a More Concrete Understanding of Metering, Legitimacy and Revenue Flows," *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain Beyond*, Nov. 2022. doi: 10.6084/m9.figshare.21534360.v1.
- [13] W. Xu, J. Li, M. Dehghani, and M. GhasemiGarpachi, "Blockchain-based secure energy policy and management of renewable-based smart microgrids," *Sustainable Cities and Society*, vol. 72, p. 103010, 2021.
- [14] A. Sims, "Blockchain and decentralised autonomous organisations (daos): The evolution of companies?" *28 New Zealand Universities Law Review*, 2019.
- [15] J. Howell, "Top decentralized autonomous organization (DAO) use cases examples." (2022), [Online]. Available: <https://101blockchains.com/top-dao-use-cases-and-examples/>.
- [16] S. Hassan and P. De Filippi, "Decentralized autonomous organization," *Internet Policy Review*, vol. 10, no. 2, pp. 1–10, 2021.
- [17] U. DAO, "Uniswap exchange." (2022), [Online]. Available: <https://uniswap.org/>.
- [18] lexDAO: Building the next generation of contracts. "Uniswap exchange." (2022), [Online]. Available: <https://www.lexdao.coop/>.

- [19] H. Rao Bokkissam, I. Savelli, T. Morstyn, and P. Cuffe, "Towards a distributed autonomous organisation for financing, governing and disbursing revenues of a battery energy storage system," *iGET blockchain Conference*, 2022.
- [20] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, 2014.
- [21] E. Foundation. "Solidity." (2022), [Online]. Available: <https://docs.soliditylang.org/en/latest/index.html>.
- [22] N. Foundation. "Hardhat framework." (2022), [Online]. Available: <https://hardhat.org/hardhat-runner/docs/getting-started>.
- [23] REMIX. "Remix integrated development environment." (2022), [Online]. Available: <https://remix-project.org/>.
- [24] O. Foundation. "Openzeppelin contracts." (2022), [Online]. Available: <https://openzeppelin.com>.
- [25] K. Lifanova. "Solidity override vs. virtual functions." (2020), [Online]. Available: <https://medium.com/upstate-interactive/solidity-override-vs-virtual-functions-c0a5dfb83aaf>.
- [26] "Transaction for purchasing dao tokens." (2022), [Online]. Available: <https://goerli.etherscan.io/tx/0x8b8d0e3afab14a7ee21545ca01986260f8902d5538e28c68dedd3cef1818b667>.
- [27] "Transaction for creating a proposal for charging method." (2022), [Online]. Available: <https://goerli.etherscan.io/tx/0x98bda56c3fcbca54c17fab8547f3bcb81a854cf577296eca3942bfc6f8a5897e>.
- [28] "Transaction for voting on a proposal." (2022), [Online]. Available: <https://goerli.etherscan.io/tx/0x1bbdca8b99f522aad0f1ea313fecaf3ac3dde4795dd4a6ec705046539e4454e7>.
- [29] "Transaction for creating a proposal to release funds." (2022), [Online]. Available: <https://goerli.etherscan.io/tx/0x661b90acbb44366e392aad677200d019879c96a80fa4f0f5a78ead15d934d161>.
- [30] "Transaction for withdrawing revenue from the smart contract." (2022), [Online]. Available: <https://goerli.etherscan.io/tx/0xd764c77052915deebb0bf95d0b69a615f5959bb4e02f9367425315969f97ae57>.