

Article

Towards Blockchain-Based Energy Trading: A Smart Contract Implementation of Energy Double Auction and Spinning Reserve Trading

Uyikumhe Damisa ^{1,2,*} , Nnamdi I. Nwulu ^{1,2}  and Pierluigi Siano ^{2,3} 

¹ Center for Cyber Physical Food, Energy and Water Systems (CCP-FEWS), University of Johannesburg, Johannesburg 2092, South Africa; nnwulu@uj.ac.za

² Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa; psiano@unisa.it

³ Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano, Italy

* Correspondence: uyikumhe@yahoo.com

Abstract: The decentralization of power generation driven by the rise in the adoption of distributed energy resources paves the way for a new paradigm in grid operations. P2P energy trading is beneficial to the grid as well as the connected peers. A blockchain-based smart contract is well suited to transparently facilitate trades between energy consumers and producers without the services of intermediaries. In this paper, Ethereum-based smart contracts that facilitate double energy auction and spinning reserve trading are developed with Solidity, compiled, and deployed within the Remix IDE. Willing energy sellers/buyers submit offers/bids to a contract that implements the double auction procedure. In order to fulfil energy supply obligations, sellers are also able to purchase spinning reserves via another smart contract. The smart contracts' effectiveness in performing the auction procedure and making payments is confirmed using an energy/reserve market scenario. The proposed scheme encourages further adoption of distributed energy resources and participation in local P2P energy trading.

Keywords: blockchain; double auction; P2P energy trading; spinning reserve; smart contract



Citation: Damisa, U.; Nwulu, N.I.; Siano, P. Towards Blockchain-Based Energy Trading: A Smart Contract Implementation of Energy Double Auction and Spinning Reserve Trading. *Energies* **2022**, *15*, 4084. <https://doi.org/10.3390/en15114084>

Academic Editors: Zhengmao Li, Tianyang Zhao, Ke Peng, Jinyu Wang, Zao Tang and Sumedha Sharma

Received: 7 April 2022

Accepted: 12 May 2022

Published: 1 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The increased adoption of renewable energy resources at the bulk generation level as well as the distribution level, the rise in demand-side participation, and the advent of new technological innovations are driving significant changes in the electricity industry [1]. An aspect of the industry that is being significantly impacted is energy trading. It originally involved only large players in the industry. However, household prosumers are now able to sell excess energy generated. Traditionally, centralized systems are employed to facilitate such energy transactions, but this results in high transaction costs and management inefficiencies [2]. Moreover, centralized systems are susceptible to a single point of failure, censorship, hacking, and privacy issues, among others. Peer-to-peer (P2P) systems have proven effective in facilitating transactions between peers in a network without the need for centralized authorities. In a P2P environment, electricity consumers with onsite generators can sell excess energy to neighbours in need, thereby forming local energy markets. P2P energy trading, which involves the local matching of energy supply and demand, offers economic and secure load balancing in a power grid with intermittent renewable energy sources and variable demand [3]. Real-world P2P energy trading schemes have already been implemented. Piclo is a platform that facilitates renewable energy trading. Sonnen Community is a P2P energy trading platform that allows the sharing of surplus energy among members of a virtual community. In the Netherlands, a similar platform exists, called Vandebbron. A number of these existing P2P energy trading platforms tend

to employ traditional centralized database technologies, which results in issues regarding data privacy and transaction clumsiness [2]. Centralized systems are not easily scalable when there is a rapid rise in the number of prosumers [4]. A prominent technology being investigated for the implementation of secure, transparent P2P transactions is blockchain. The incorporation of blockchains into energy trading systems offers low-cost high computing power and optimal solutions via consensus mechanisms [5]. The technology rose to the limelight after the successful creation of Bitcoin—a global digital currency that is underpinned by it. Peers on the bitcoin network—a decentralized P2P network—can send and receive bitcoins. A blockchain is a cryptographically secured distributed ledger of records of P2P transactions on a P2P network. With this ledger, digital transactions can proceed without the services of a third-party organization [6]. Blockchain-based P2P energy trading networks have been successfully implemented in real life, one of such being the popular Brooklyn microgrid powered by Exergy. It is a local energy market that allows residential and commercial solar prosumers to sell excess solar generation to neighbours in New York City. Research in blockchain-based energy trading is gradually gaining traction, with researchers approaching it from various perspectives. The authors of [7] proposed an Industrial-Internet-of-Things (IIOT)-assisted energy trading scheme that is based on blockchain. Their scheme protects participants' electricity information whilst ensuring supply and demand balance in the electricity market. A suitable blockchain-based P2P energy transactions framework for typical IIOT transaction scenarios was developed in [8].

While transactions may be public over a blockchain-based P2P network, the privacy and security of peers can remain uncompromised [9]. In the blockchain-based energy trading framework proposed in [10], users' privacy is preserved; moreover, peers on the network are encouraged to trade with near neighbours by factoring in the electrical distance between peers into the pricing mechanism. A reputation factor, which is based on the past behaviour towards fulfilling energy commitments, is associated with participants on the network. A new double-layered blockchain-based energy trading platform was developed in [11]. A market layer features the auction and clearing mechanism, while a blockchain layer offers security and automated real-time settlements.

In [12], the use of blockchains and smart contracts was extended beyond energy trading to incorporate carbon allowance trading.

As demonstrated in [13], blockchain can as well be employed for energy management. The paper proposed an integrated platform for energy management along with a bilateral trade mechanism implementation. The authors of [14] have also incorporated the use of blockchain and smart contracts in the management of a distributed energy environment. In [15], the optimal power flow problem was tackled in a decentralized manner with the aid of a blockchain-based consensus algorithm.

With the smart contract functionality, the versatility of blockchains is greatly enhanced. Smart contracts based on blockchain possess the potential to facilitate credible transactions in the absence of third parties [2]. They strictly execute rules regarding payment and trading without human intervention, hence improving fairness and security in energy trading [2]. In [16], smart contracts were used to automate the bidding procedure for transactions for demand and supply of energy in smart cities. The authors of [17] incorporated blockchain and smart contracts in their proposed decentralized microgrid electricity market mechanism.

Table 1 reviews some related works in the literature, and a comparative study of the Ethereum and Hyperledger Fabric blockchains can be found in [18]. The work in [19] compares various other distributed ledger technologies suitable for energy transactions.

In [11], the investigated grid is partitioned into microgrids, and multi-level trading is proposed. Energy surplus/deficit resulting from intra microgrid trades are traded at the inter microgrid trade level. In [2] as well, a distributed system operator is responsible for handling energy imbalances. In contrast with these papers, we propose a smart contract-based scheme where spinning energy is provided by peers within the network to cater for energy deficits resulting from the failure of suppliers to meet their obligations. In

other words, energy producers (some of which depend on unpredictable renewable energy sources) make offers in terms of quantity and cost of energy and are expected to deliver if successfully matched with bidders. Should there be a shortfall in energy production at the expected time of delivery, the producer is able to source for the energy deficit from network peers (via the same P2P scheme) to meet supply obligations. The scheme encourages grid independence and is hence suitable for standalone grids. Ethereum-based smart contracts that facilitate energy double auction and spinning reserve trades are proposed. For the double energy auction, sellers/buyers submit offers/bids in terms of quantity and per-unit cost, and according to its auction mechanism, a smart contract performs the matching of buyers and sellers as well as payments and refunds for energy transactions. Another smart contract receives spinning reserve offers and demand from buyers and settles reserve-related payments. The contracts are developed, compiled, deployed, and tested within the Remix integrated development environment.

Table 1. Objectives, blockchain networks and findings of selected articles.

Ref.	Objective	Blockchain Platform	Findings
[2]	To present a general blockchain-based framework that facilitates P2P trading in retail electricity markets	Ethereum private chain	The platform was observed to be capable of achieving efficient and effective transactions among multiple players
[12]	To propose a blockchain-based P2P framework that facilitates the trading of energy and carbon allowance	Ethereum	The proposed scheme is seen to outperform both centralised or aggregator-based trading, in terms of energy management and carbon emissions
[20]	To present an hour ahead blockchain-based trading network and a comparison of auction mechanisms	Hyperledger Fabric	Regardless of auction mechanisms, the price-only game-theoretical bidding strategy results in near-ideal economic efficiency
[21]	To propose new approaches to determining the bilateral trading preferences of residences in a local P2P energy market	Hyperledger Fabric	For both of the strategies proposed, compared to a baseline case, the cost of procuring energy and grid interaction of all P2P trading peers is reduced
[22]	To design a platform for P2P energy trading based on blockchain	Ethereum	Using the proposed platform, distant users are able to perform reliable P2P transactions without centralized authority intervention
[23]	To implement a hybrid P2P energy trading market employing blockchain	Ethereum	Customers' electricity cost reduction was achieved
[Present paper]	To develop a smart contract-based P2P scheme for double energy auction and reserve trading	Remix IDE (Ethereum)	Smart contracts are effective in performing energy and reserve trading, as well as making payments

The rest of this paper is organized as follows. An overview of the concepts of blockchain and smart contracts is provided in Section 2, followed by a description of a smart contract-enabled P2P energy trading framework in Section 3. The proposed trading mechanism programmed into the smart contracts is detailed in Section 4, and Section 5 shows the case study considered. The results obtained from tests conducted are reported and discussed in Section 6, and the paper concludes with Section 7.

2. Overview of Blockchain and Smart Contracts

A blockchain is a distributed ledger system in which P2P transactions are recorded on a ledger that is shared among key network peers. A combination of cryptography and consensus mechanisms helps to secure blockchain networks. Intermediaries can be eliminated with blockchains, thereby improving the efficiency of systems; peers are empowered to securely transact with one another without the need for central authorities. While the bitcoin blockchain brought the technology to the limelight, it is limited in its scope of application, being originally intended to facilitate the P2P exchange of the bitcoin

cryptocurrency. The Ethereum blockchain improves upon that of Bitcoin in that it offers smart contract functionality. This feature extends the application of blockchains way beyond P2P electronic payments. A smart contract is a piece of a computer program that is hosted by and executed on a blockchain. The Ethereum white paper [24] describes smart contracts as “systems which automatically move digital assets according to arbitrary pre-specified rules”. It has the potential to replace many third-party roles in industries. A traditional contract between parties can be transformed into computer code and enforced by a blockchain. Being powered by blockchain technology, smart contracts are immutable, cryptographically secured, and trustless. With the advent of smart contracts, blockchains such as Ethereum and other smart contract-hosting blockchains find applications in a wide range of areas. The Ethereum blockchain hosts the highest number of smart contracts compared to other blockchain platforms. Smart contracts in Ethereum are written mainly using the Solidity programming language. To facilitate the easy development and testing of smart contracts on the Ethereum blockchain, various freely available tools have been developed, one of such being the Remix browser-based platform [25]. Remix is an integrated development environment that provides facilities for smart contract development, testing, compilation, and deployment. The environment simulates an entire blockchain setup that can be fully accessed via a browser. It supports smart contract development using one of the two Ethereum smart contract programming languages—Solidity and Vyper. Contracts may be deployed within the browser to an injected Web3 or a Web3 provider. Test Ethereum accounts loaded with fake ethers that can be used to pay for transaction fees are provided in Remix. The ethers can also be transferred among these accounts, for instance, while testing a smart contract functionality. In general, the interactions between these accounts mimic those obtainable between real nodes on the Ethereum blockchain.

3. Smart Contract-Based Peer-to-Peer Energy Trading Framework

The P2P energy trading system presented is inspired by that reported in [12]. Each peer within the grid has an onsite smart meter that is connected to a smart contract. Peers can communicate with the smart contract via a mobile or desktop application. The auction mechanism that is used to match sellers and buyers is programmed into this smart contract. The contract receives offers/bids from sellers/buyers and matches them according to its auction mechanism. While sellers simply submit offers to the contract, buyers submit bids as well as commit funds commensurate with the bids they make. The funds are held by the smart contract, and sellers are paid by the contract for the energy delivered. Figure 1 depicts a simplified smart contract-based P2P energy trading system.

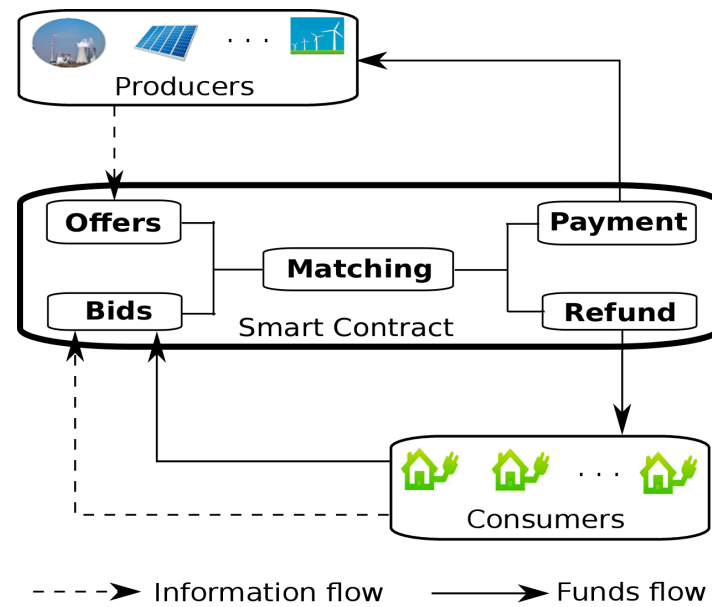


Figure 1. Smart contract-based P2P energy trading scheme.

4. Proposed Smart Contract-Based Energy and Reserve Trading Mechanism

The proposed energy double auction and reserve trading scheme is programmed into smart contracts according to Algorithms 1 and 2. Figures 2 and 3 provide systematic diagrams of each contract, and flowcharts of the contracts' logic are presented in Figures 4 and 5. The following describes the various functions contained in the contracts:

Algorithm 1 Smart contract-based energy double auction mechanism

```

F1: Constructor
    //Executed at contract deployment
    //Assigns the address of the contract deployer to a variable "owner"
F2: setRate
    input: Ether price in $
    require: function caller = owner
    rate = Ether price
F3: CheckEtherEquiv
    input: Energy price in $/MWh
    output: Energy price * 1018/rate
F4: offers
    input: Seller's energy offer (SEO) in MWh; Seller's offer price (SOP) in $
    store in 'sellers' mapping
F5: verifiedOffers
    input: account address of offeror; (SEO); (SOP)
    require: SEO & SOP in 'sellers' match, for all account addresses.
    populate a new mapping 'vSellers' with the newly ordered account addresses, SEOs & SOPs
    create an array each for account addresses, SEOs & SOPs
    remove data from 'sellers' mapping
F6: bids
    require: bid price is equal to the amount of ether sent multiplied by rate
    input: buyer's energy bid (BEB) in MWh; buyer's bid price (BBP) in $
    store in 'buyers' mapping
    //The 'buyers' mapping is sent out for re-ordering in descending order of BBPs then the new list
    is sent back to the smart contract for verification. Re-ordering within the smart contract is
    expensive.
F7: verifiedBids
    input: account address of bidder; BEB and BBP
    require: BEB and BBP in 'buyers' match, for all account addresses.
    populate a new mapping with the newly ordered account addresses, BEBs and BBPs
    create an array each for account addresses, BEB and BBP
    remove data from 'buyers' mapping
F8: matching
    require: function caller = owner
    for each seller
        x = 0; yy = 0; pay = 0;
    for each buyer
        if BBP < SOP, x = 1; break out of loop;
        if BEB = 0, continue to next buyer
        if SEO < BEB
            BEB = BEB - SEO
            pay = pay + (BBP + SOP) * SEO * 1018/(2*rate) }   SEO = 0; break out of loop;
        else
            SEO = SEO - BEB
            pay = pay + (BBP + SOP) * BEB * 1018/(2*rate)
            BEB = 0
        if last buyer, yy = 1; break out of loop;
    end for

```

Algorithm 1 *Cont.*

```

    if  $x > 0$ , break out of loop;
    playlist.push(pay)
    if  $yy > 0$ , break out of loop;
    end for
F9: paySellers
    require: function caller = owner
    for each element in array
        pay corresponding seller
    end for
F10: refundBuyers
    require: function caller = owner
    for each bidder,
        if unsuccessful
            refund committed funds
    end for

```

Algorithm 2 Smart contract-based reserve trading mechanism

```

F1: Constructor
    //Executed at contract deployment
    //Assigns the address of the contract deployer to a variable "owner"
F2: setRate
    input: Ether price in $
    require: function caller = owner
    rate = Ether price
F3: SROffers
    input: Spinning reserve offers (SRO) in MWh; Spinning reserve price (SRP) in $
    store in 'sRsellers' mapping
F4: verSROffers
    input: account address of spinning reserve seller; (SRO); (SRP)
    require: SRO & SRP in 'sRsellers' match, for all account addresses.
    populate a new mapping 'sRvSellers' with the newly ordered account addresses, SROs & SRPs
    create an array each for account addresses, SROs & SRPs
    remove data from 'sRsellers' mapping
F5: vSReserve
    sum1 = 0;
    for each SRO
        if  $SRO = 0$ , continue to next
        if  $SRO < \text{reserve demand (RD)}$ 
             $RD = RD - SRO$ 
            store SRO address and revenue
             $sum1 = sum1 + \text{revenue}$ 
             $SRO = 0$ 
        else
             $SRO = SRO - RD$ 
            store SRO address and revenue
             $sum1 = sum1 + \text{revenue}$ 
             $RD = 0$ 
            store buyer's address along with sum1 in "sRbuyers"
            store sum1 in a public variable
        end for
F6: pay4SR
    Reserve buyers pay for the reserve amount requested
F7: settleSR
    require: function caller = owner
    pay all succesful reserve sellers

```

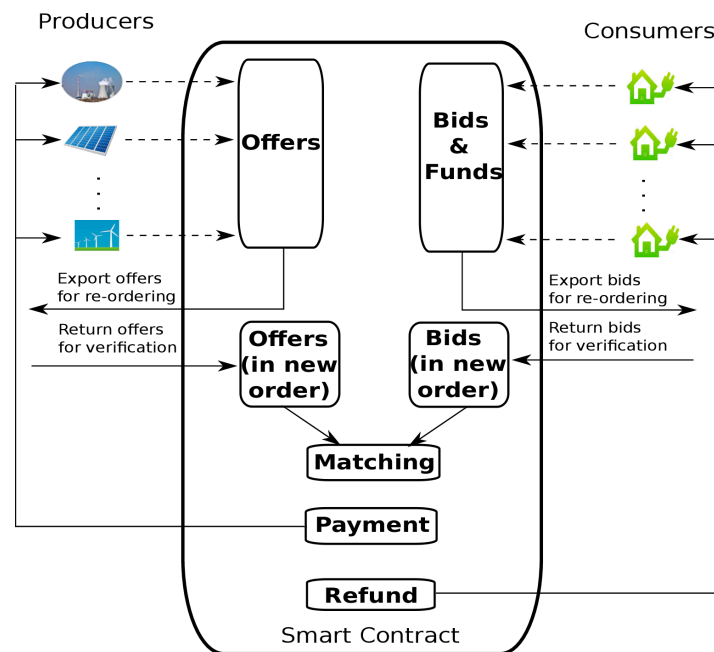


Figure 2. Systematic smart contract-based double auction P2P energy trading scheme (smart contract A). Energy sellers/buyers submit their offer/bid to the smart contract (F4/F6). Buyers make payment to the contract while submitting bid (F6). Offers/bids are sent outside the contract for re-ordering, and sent back into the contract (F5/F7) which then matches sellers and buyers using a double auction mechanism (F8). The contract pays successful sellers and refunds unsuccessful bidders (F9/F10).

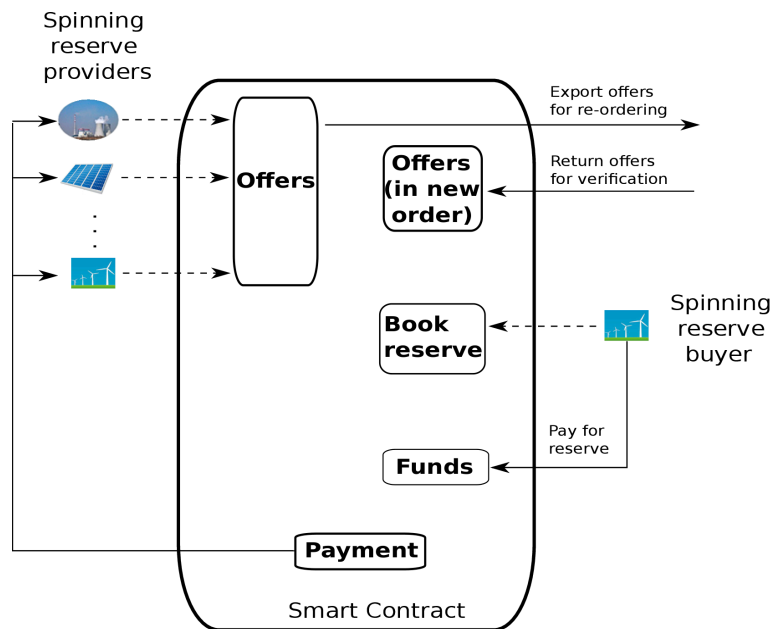


Figure 3. Systematic smart contract-based reserve trading scheme (smart contract B). Reserve sellers submit their offer to the smart contract (F3). Offers are sent out to be re-ordered and returned to the contract (F4). Buyers submit their demand (F5), as well as make payment (F6), to the smart contract which then pays successful sellers (F7).

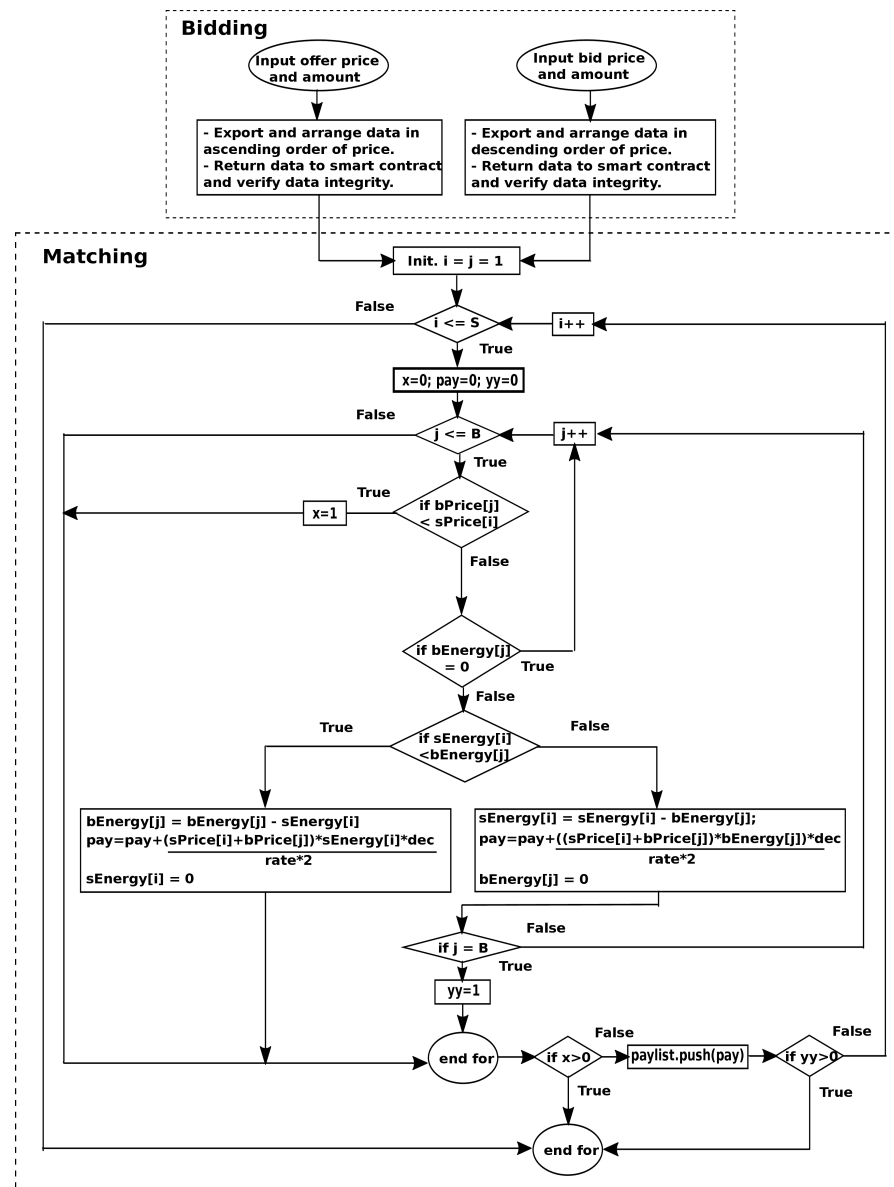


Figure 4. Flowchart of the matching function in the smart contract-enabled energy double auction scheme. The ‘Bidding’ section involves the reception and re-ordering of offers and bids, while ‘Matching’ uses a double-auction mechanism to determine the amount and cost of energy to be traded between sellers and buyers. See Algorithm 1 (sEnergy[i]—quantity of energy being offered by seller i—bEnergy[j]—quantity of energy being bid for by buyer j—bPrice[j]—bid price of buyer j—sPrice[i]—offer price of seller I—rate—dollar value of ether—S—number of sellers—B—number of buyers—paylist—array of sellers’ pay).

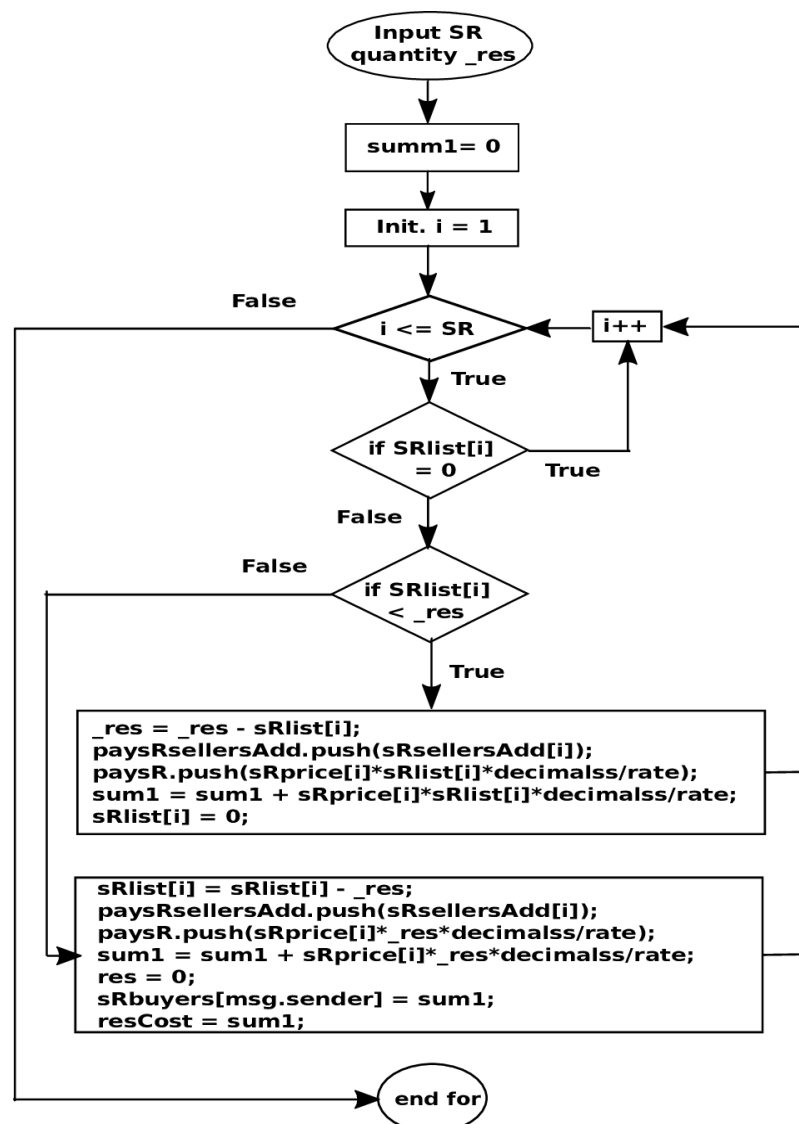


Figure 5. Flowchart of the reserve trading function of the smart contract-enabled spinning reserve trading scheme. It depicts the logic behind the smart contract’s determination of the cost of reserve demanded based on the availability (and offer price) of seller(s) to meet a buyer’s demand. See Algorithm 2 ($_res$ —quantity of SR— $resCost$ —total cost of SR—SR—number of SR offers— $sRlist[i]$ —SR seller i ’s offer quantity— $sRprice[i]$ —SR seller i ’s offer price— $sRbuyers$ —mapping of SR buyers and cost).

4.1. Smart Contract A

Function 1, referred to as a Constructor, self-executes only upon deployment of the smart contract. It can be used to restrict access to other functions within the smart contract. In the present case, the account address of the contract deployer is assigned to a variable “owner”, which is later used within the smart contract to restrict access to certain sensitive functions.

Function 2 enables the prevailing value of ether in dollars to be set. It can only be called by the “owner”. The function is called with the current value of ether (the native cryptocurrency of the Ethereum blockchain) as input which it then stores in the variable “rate”.

Function 3 helps prospective energy buyers determine the ether equivalent of their bid prices. This is important since it is the currency being used to facilitate energy trading on the platform.

Willing sellers call Function 4 to submit their offer quantities and costs, which are then stored in the Solidity mapping named “sellers”. It is then sent out for re-ordering in ascending order of prices and sent back to the smart contract for verification. This re-ordering process is a necessary step of the auction procedure and is performed outside the smart contract to reduce the gas fees (fees paid for initiating transactions on the blockchain) incurred for transactions within the smart contract.

The re-ordered off-chain data of sellers’ account addresses, sellers’ energy offers (SEOs), and sellers’ offer prices (SOPs) are received by Function 5, compared with the original on-chain data contained in ‘sellers’ to ensure data integrity and then stored in a new mapping ‘vSellers’. Arrays of sellers’ account addresses, SEOs, and SOPs, all in the new order, are created for use in other functions. In order to avoid data duplication, each verified entry is deleted from the original “sellers” mapping.

Function 6 is similar to Function 4, except that buyers are required to commit an amount of ether equivalent to the total cost of energy being bid for. This amount is stored in the smart contract until the auction is ended, after which appropriate payments are made to sellers.

Function 7 is similar to Function 5, except that it is for buyers’ bids.

Function 8 performs the matching of sellers and buyers, as well as the determination of the clearing price for each matched pair. This price is derived by computing the mid-point between a seller’s offer price and a buyer’s bid price [26].

Upon successful delivery of energy by sellers, Function 9 is called to initiate payment to the sellers.

Bidders who were partly or completely unsuccessful obtain a refund when Function 10 is called.

4.2. Smart Contract B

Function 1, referred to as a Constructor, self-executes only upon deployment of the smart contract.

Function 2 enables the prevailing dollar value of ether to be set.

Function 3 is called by spinning reserve (SR) sellers to submit their offers which are sent out to be arranged in ascending order of SR cost. The data are returned to Function 4, which checks to ensure data integrity.

Prospective SR buyers call Function 5 with the requested quantity of SR as input, the cost of which is passed to a public variable that can be viewed by the buyer. The cost viewed is paid to the smart contract using Function 6, and successful reserve sellers are paid when Function 7 is called. It is assumed that prospective buyers who call Function 5 call Function 6 as well.

It should be noted that the manual calls to some of these functions can be automated in the practical implementation of the scheme.

5. Simulation Setup and Case Study

The Remix IDE is employed in this study. It is an environment that simulates a blockchain network of peers/nodes. Remix provides 15 unique nodes, each equipped with 100 ethers (which can be used to pay for goods, services, or transaction fees). For the simulation in this study, a node is taken as the administrator and performs crucial functions such as initiating payments. Energy and spinning reserve sellers/buyers are each assigned a unique node from the 14 other nodes. Hence, a simulated blockchain network of sellers and buyers who can transact in a P2P manner is formed. These entities are then able to interact with the smart contracts developed in the environment. In our simulation, offers/bids are manually submitted to the smart contracts using each entity’s node, and the administrator node is used to perform administrative functions (such as payments and refunds). Solidity is used to write the contract codes, and a PC with an Intel Core i5-6200U processor, memory 7.7 GiB, running Ubuntu 20.04.1 LTS operating system is used to perform the simulations.

For efficiency and economic reasons, a small grid is more suitable for the proposed scheme. Bid/offer submissions from the buyers/sellers, as well as payments/refunds made to sellers/buyers, each incur a transaction fee. Hence, a high aggregate transaction fee is incurred for a grid with many entities. Moreover, excessive transactions over the blockchain network may result in a spike in transaction fees or low transaction speeds. Therefore, to apply the scheme to a large-scale grid, offers/bids should be aggregated. By employing an appropriate aggregation mechanism, the aggregate transaction cost for a large system can be reduced, and efficiency improved.

An example of eight generators and eight large customers with offers/bids data shown in Table 2 is obtained from [27] and adopted in this study. It is suitable for use as it provides a convenient way to verify the accuracy of the energy auction logic, reserve trading procedure, and payments effected by the proposed smart contracts. An exchange rate of 1\$ to 6.44 Yuan has been used to convert the original data from Yuan/kWh to \$/kWh. Table 3 is derived from the results obtained from the auction procedure performed on the data in Table 2.

Table 2. Sellers/Buyers' offer/bid amount and prices [27].

Seller	Offer (\$/MWh)	Quantity (MWh)	Buyer	Bid (\$/MWh)	Quantity (MWh)
A	65	2000	a	64	3000
B	64	3000	b	64	3000
C	62	5000	c	65	2000
D	66	1000	d	63	4100
E	65	2000	e	62	5000
F	63	4000	f	65	2000
G	62	5000	g	66	1000

Table 3. Spinning reserve data.

Seller	Offer (\$/MWh)	Quantity (MWh)	Buyer	Quantity (MWh)
A	78	1600	C	1500
B	77	2400	F	500
D	79	800	G	1000
E	78	1600		

6. Results and Discussion

Figure 6 depicts the offering and bidding stage, along with the matching of offers and bids achieved by the Smart Contract A. Sellers' offerings and buyers' bids in terms of energy amount and unit cost are submitted to the smart contract as shown in the figure. Based on the double auction procedure programmed into the contract, buyers are matched with sellers, as shown. The procedure involves re-arranging sellers and buyers in ascending and descending order of unit cost, respectively, resulting in the following orders: sellers: G, C, F, B, A, E, and D and buyers: g, f, c, a, b, d, and e. Matching, therefore, commences from Seller G and Buyer g, and the clearing price is set at the midpoint of each matched pair's offer/bid prices. A key criterion for matching pairs is the buyer's bid price exceeds the seller's offer price. In the case study, Seller G's energy offer is exhausted by being matched with buyers g, f and c. Following Seller G on the ordered sellers' list is Seller C, who is then matched with buyers a and b. Finally, Seller F's offer quantity is exhausted on buyers b and d, bringing the auction round to a close, as buyer d's bid price is lower than Seller B's offer price.

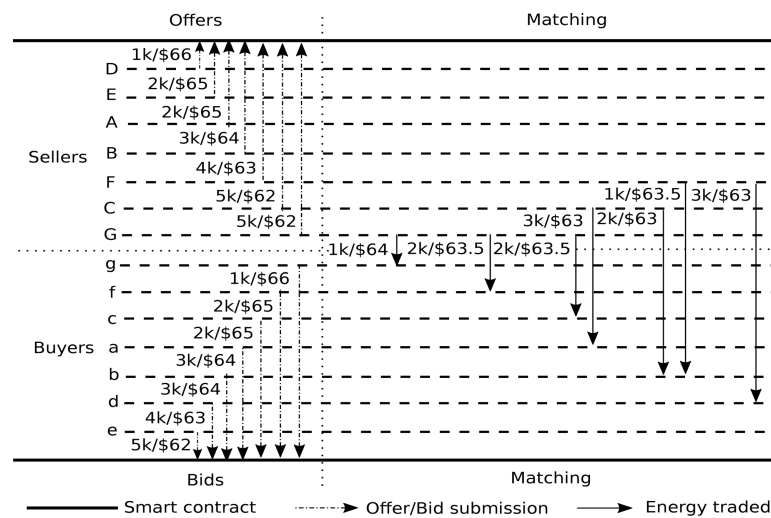


Figure 6. Smart contract execution of energy double auction. Energy sellers/buyers submit their offer/bid (amount of energy in MWh and cost per unit in \$/MWh) to the smart contract which matches sellers and buyers using a double auction mechanism. The matching process determines the amount and cost of energy to be exchange between seller/buyer pairs.

Figure 7 shows the flow of funds between the various entities involved in the scheme. While making bids, buyers are mandated to commit funds commensurate with their energy requirements. These funds, which are in Ethers, are held by the smart contract until energy delivery is confirmed, thereby exploiting the escrow capability of smart contracts. Upon confirmation of energy delivery, the “paySellers” function within the contract is called upon to send funds to the successful energy sellers. The “refundbuyers” are also called to refund unsuccessful bidders who, as required, committed funds. In Figure 6, it is apparent that sellers G and C made similar energy offers in terms of quantity and price, sold all of them, and hence should receive similar revenues. However, a slight discrepancy in their revenues can be seen in Figure 7 due to their positions on the ordered sellers’ list. While the order in which offers/bids were received can be used to decide their place in the ordered list, this might not always be a fair basis. This issue is, therefore, worth further investigation to determine the appropriate order for similar offers/bids, especially when the order of receiving offers/bids is unavailable or considered irrelevant. Similar trading schemes also need to be aware of this issue.

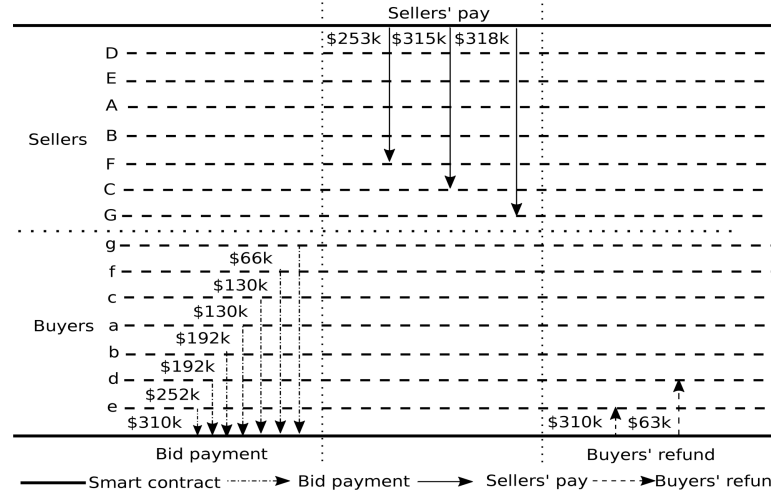


Figure 7. Smart contract execution of energy double auction payments. Energy buyers make payment along with their bids, to the smart contract. The contract pays successful sellers and refunds unsuccessful bidders.

From the discussion above, the matching of energy sellers and buyers under the double auction procedure carried out by the smart contract can be seen to be logically correct (based on the Case Study data). The same accuracy is also seen under the reserve trading scheme. In addition, the payments made by the contract to the various entities can be observed to be arithmetically accurate. These observations ascertain the effectiveness of smart contracts.

The foregoing simulation results assume that sellers indeed deliver their energy offers. This would be unrealistic, especially with renewable energy-based sellers. A reserve trading scheme based on the previous simulation is further considered in this study. It is assumed that after the auction round, unsuccessful sellers (A, B, D and E) become spinning reserve sellers. However, due to output variability, they can only guarantee the availability of 80% of their initial offering, which will be offered at about 20% higher than the original offering. The successful sellers, on the other hand, (G, C and F) are assumed to have experienced shortfalls in available energy and hence need additional energy from the reserve market to fulfil their energy supply obligations. This assumes that sellers are obliged to fully meet their energy offers. These are shown in Table 3. The results of the spinning reserve trades are depicted in Figure 8. As with the energy auction scheme, spinning reserve offers are submitted to Smart Contract B, as well as the demands for it. However, demands are treated in near-real-time as the contract receives them. Since sellers are obliged to meet their supply commitments, they need to be wary of the quantity they offer, as failing to meet up would result in the need to purchase an expensive spinning reserve. Alternative methods of handling such failure to meet supply commitments are open for further investigation. For instance, buyers could have the option of sourcing for energy to cater to their energy deficit while defaulting sellers are penalized. In a microgrid, transactive controllers may be employed to purchase SR capacities needed by the grid, and appropriate payment schemes developed. The cost of smart contract computations increases with the number of transacting assets [28]. Hence, this should not be neglected. In the case of renewable energy assets, uncertainties in power production could be further considered, as suggested in [29]. To address uncertainties, a blockchain-based demand response scheme (similar to that presented in [30]) may be adopted.

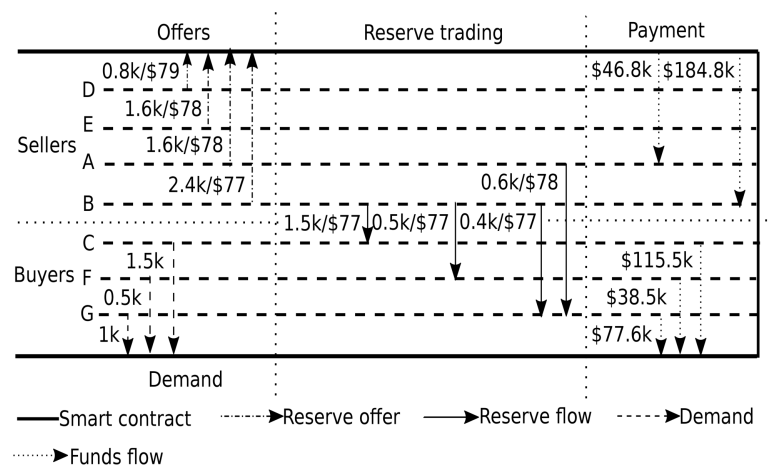


Figure 8. Smart contract execution of reserve trades and payments. Reserve sellers submit their offer (amount of reserve in MWh and cost per unit in \$/MWh) to the smart contract. Buyers submit their demand, as well as make payment, to the smart contract, which matches sellers and buyers. The contract then pays successful sellers.

In contrast to the present study’s proposal of Ethereum—a public blockchain—a few relevant works in the literature [20,21] utilize Hyperledger Fabric—a private blockchain. Private blockchains are typically faster than public ones. However, they can be somewhat centralized and less secure [31]. The proof-of-work consensus mechanism being employed

by Ethereum, coupled with its mass adoption, makes it quite secure. The blockchain is also more decentralized than typical private blockchains [31]. Although data privacy is a natural feature of private blockchains, transactions over Ethereum are inherently transparent (users are, however, anonymous). Furthermore, being at the forefront of the smart contract innovation, Ethereum enjoys the first-mover advantage; it hosts more decentralised applications than most of its counterparts. Ethereum-based smart contracts can also be adapted for deployment on other blockchains such as Binance Smart Chain, Avalanche, Polygon, etc. Some other relevant works [12,22,23] have employed Ethereum; however, the present work incorporates the concept of P2P spinning reserve trading among network peers.

7. Conclusions

Smart contracts possess enormous potential within and beyond the energy sector. Ethereum smart contracts have been successfully adopted in various sectors. The features of smart contracts make them a natural fit for facilitating P2P energy trading. In this paper, Ethereum-based smart contracts that enable energy auctions and SR trading were proposed. Energy sellers and buyers submit offers and bids respectively to a smart contract that performs a double auction procedure. Sellers who fail to meet their energy supply obligations can purchase SR capacities from other peers via another smart contract to cater for supply shortfalls. The contracts were written in Solidity and deployed in the Remix IDE. In order to test the proposed smart contracts, data for an energy market scenario were adopted. The results of the investigation validate the effectiveness of the proposed smart contracts in facilitating double auction energy trade, reserve trades, and making payments.

Future work could take into account the cost incurred by initiating on-chain transactions from the smart contract. This could be fairly distributed among participating peers of the scheme so that these peers jointly bear the financial burden of running the system. Additionally, since buyers commit funds at the time of bidding (when the actual clearing cost of energy is unknown), committed funds would usually differ from the clearing cost of energy, leaving the excess funds in the smart contract. This issue can be addressed as an advancement of the present work. Finally, the proposed scheme can be adapted for use in an auction-based energy storage sharing setting as future work. Distributed storage units may be pooled to offer ancillary services to utilities, with aggregate units and utilities being matched by a smart-contract-based auction procedure and payments made by the contract as well.

Author Contributions: Conceptualization, U.D.; methodology, U.D.; software, U.D.; validation, U.D.; investigation, U.D.; writing—original draft preparation, U.D.; writing—review and editing, U.D., N.I.N. and P.S.; supervision, N.I.N.; project administration, N.I.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rahimi, F.; Ipakchi, A.; Fletcher, F. The changing electrical landscape: End-to-end power system operation under the transactive energy paradigm. *IEEE Power Energy Mag.* **2016**, *14*, 52–62. [[CrossRef](#)]
2. Han, D.; Zhang, C.; Ping, J.; Yan, Z. Smart contract architecture for decentralized energy trading and management based on blockchains. *Energy* **2020**, *199*, 117417. [[CrossRef](#)]
3. Li, J.; Ye, Y.; Strbac, G. Stabilizing peer-to-peer energy trading in prosumer coalition through computational efficient pricing. *Electr. Power Syst. Res.* **2020**, *189*, 106764. [[CrossRef](#)]

4. Siano, P.; de Marco, G.; Rolan, A.; Loia, V. A survey and evaluation of the potentials of distributed ledger technology for peer-to-peer transactive energy exchanges in local energy markets. *IEEE Syst. J.* **2019**, *13*, 3454–3466. [CrossRef]
5. Wang, N.; Zhou, X.; Lu, X.; Guan, Z.; Wu, L.; Du, X.; Guizani, M. When energy trading meets blockchain in electrical power system: The state of the art. *Appl. Sci.* **2019**, *9*, 1561. [CrossRef]
6. Monroe, J.G.; Hansen, P.; Sorell, M.; Berglund, E.Z. Agent-Based Model of a Blockchain Enabled Peer-to-Peer Energy Market: Application for a Neighborhood Trial in Perth, Australia. *Smart Cities* **2020**, *3*, 1072–1099. [CrossRef]
7. Guan, Z.; Lu, X.; Wang, N.; Wu, J.; Du, X.; Guizani, M. Towards secure and efficient energy trading in IIoT-enabled energy internet: A blockchain approach. *Future Gener. Comput. Syst.* **2020**, *110*, 686–695. [CrossRef]
8. Hu, W.; Li, H. A blockchain-based secure transaction model for distributed energy in Industrial Internet of Things. *Alex. Eng. J.* **2021**, *60*, 491–500. [CrossRef]
9. Abdella, J.; Shuaib, K. Peer to peer distributed energy trading in smart grids: A survey. *Energies* **2018**, *11*, 1560. [CrossRef]
10. Khorasany, M.; Dorri, A.; Razzaghi, R.; Jurdak, R. Lightweight blockchain framework for location-aware peer-to-peer energy trading. *Int. J. Electr. Power Energy Syst.* **2020**, *127*, 106610. [CrossRef]
11. Esmat, A.; de Vos, M.; Ghiassi-Farrokhfal, Y.; Palensky, P.; Epema, D. A novel decentralized platform for peer-to-peer energy trading market with blockchain technology. *Appl. Energy* **2021**, *282*, 116123. [CrossRef]
12. Hua, W.; Jiang, J.; Sun, H.; Wu, J. A blockchain based peer-to-peer trading framework integrating energy and carbon markets. *Appl. Energy* **2020**, *279*, 115539. [CrossRef]
13. van Leeuwen, G.; AlSkaif, T.; Gibescu, M.; van Sark, W. An integrated blockchain-based energy management platform with bilateral trading for microgrid communities. *Appl. Energy* **2020**, *263*, 114613. [CrossRef]
14. Li, Y.; Yang, W.; He, P.; Chen, C.; Wang, X. Design and management of a distributed hybrid energy system through smart contract and blockchain. *Appl. Energy* **2019**, *248*, 390–405. [CrossRef]
15. Foti, M.; Mavromatis, C.; Vavalis, M. Decentralized blockchain-based consensus for Optimal Power Flow solutions. *Appl. Energy* **2021**, *283*, 116100. [CrossRef]
16. Khattak, H.A.; Tehreem, K.; Almogren, A.; Ameer, Z.; Din, I.U.; Adnan, M. Dynamic pricing in industrial internet of things: Blockchain application for energy management in smart cities. *J. Inf. Secur. Appl.* **2020**, *55*, 102615. [CrossRef]
17. Sabounchi, M.; Wei, J. A decentralized P2P electricity market model for microgrids. In Proceedings of the 2018 IEEE Power & Energy Society General Meeting (PESGM), Portland, OR, USA, 5–10 August 2018. [CrossRef]
18. Sajana, P.; Sindhu, M.; Sethumadhavan, M. On blockchain applications: Hyperledger fabric and Ethereum. *Int. J. Pure Appl. Math.* **2018**, *118*, 2965–2970.
19. Di Silvestre, M.L.; Gallo, P.; Ippolito, M.G.; Sanseverino, E.R.; Zizzo, G. A technical approach to the energy blockchain in microgrids. *IEEE Trans. Industr. Inform.* **2018**, *14*, 4792–4803. [CrossRef]
20. Lin, J. Analysis of Blockchain-Based Smart Contracts for Peer-to-Peer Solar Electricity Transactive Markets. Master’s Dissertation, Virginia Polytechnic Institute and State University, Arlington, VA, USA, 11 December 2018.
21. AlSkaif, T.; Crespo-Vazquez, J.L.; Sekuloski, M.; van Leeuwen, G.; Catalao, J.P.S. Blockchain-based fully peer-to-peer energy trading strategies for residential energy systems. *IEEE Trans. Industr. Inform.* **2022**, *18*, 231–241. [CrossRef]
22. Kwak, S.; Lee, J. Implementation of blockchain based P2P energy trading platform. In Proceedings of the 2021 International Conference on Information Networking (ICOIN), Jeju Island, Korea, 13–16 January 2021. [CrossRef]
23. Khalid, R.; Javaid, N.; Almogren, A.; Javed, M.U.; Javaid, S.; Zuair, M. A blockchain-based load balancing in decentralized hybrid P2P energy trading market in smart grid. *IEEE Access* **2020**, *8*, 47047–47062. [CrossRef]
24. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform, White Paper. 2014. Available online: <https://nftx.com/wp-content/uploads/2021/03/EthereumWP.pdf> (accessed on 16 May 2022).
25. Readthedocs.io. Welcome to Remix’s Documentation! Remix—Ethereum IDE 1 Documentation. Available online: <https://remix-ide.readthedocs.io/en/latest/> (accessed on 29 November 2021).
26. Nicolaisen, J.; Petrov, V.; Tesfatsion, L. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Trans. Evol. Comput.* **2001**, *5*, 504–523. [CrossRef]
27. Fang, D.; Wu, J.; Tang, D. A double auction model for competitive generators and large consumers considering power transmission cost. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 880–888. [CrossRef]
28. Kirli, D.; Couraud, B.; Robu, V.; Salgado-Bravo, M.; Norbu, S.; Andoni, M.; Antonopoulos, I.; Negrete-Pincetic, M.; Flynn, D.; Kiprakis, A. Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. *Renew. Sustain. Energy Rev.* **2022**, *158*, 112013. [CrossRef]
29. Bokkisam, H.R.; Singh, S.; Acharya, R.M.; Selvan, M.P. Blockchain-based peer-to-peer transactive energy system for community microgrid with demand response management. *CSEE J. Power Energy Syst.* **2021**, *8*, 198–211.
30. Deshpande, V.; George, L.; Badis, H.; Desta, A.A. Blockchain based decentralized framework for energy demand response marketplace. In Proceedings of the NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–9.
31. Blockchain Council. Public vs. Private Blockchain: A Comprehensive Comparison. 2022. Available online: <https://www.blockchain-council.org/blockchain/public-vs-private-blockchain-a-comprehensive-comparison/> (accessed on 5 May 2022).