IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 5, 2020, accepted September 18, 2020, date of publication September 23, 2020, date of current version October 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026180

Peer-to-Peer Energy Trading in Virtual Power Plant Based on Blockchain Smart Contracts

SERKAN SEVEN[1], GANG YAO[2], (Member, IEEE), AHMET SORAN[3], (Member, IEEE), AHMET ONEN[4], (Member, IEEE), AND S. M. MUYEEN[5], (Senior Member, IEEE)

[1]Software Engineering Department, Abdullah Gül University, 38080 Kayseri, Turkey
[2]Sino-Dutch Mechatronics Engineering Department, Shanghai Maritime University, Shanghai 201306, China
[3]Computer Engineering Department, Abdullah Gül University, 38080 Kayseri, Turkey
[4]Electrical-Electronics Engineering Department, Abdullah Gül University, 38080 Kayseri, Turkey
[5]School of Electrical Engineering Computing and Mathematical Sciences, Curtin University, Perth, WA 6102, Australia

Corresponding author: Ahmet Onen (ahmet.onen@agu.edu.tr)

This work was supported by the Shanghai Science and Technology Committee Foundation under Grant 19040501700.

ABSTRACT A novel Peer-to-peer (P2P) energy trading scheme for a Virtual Power Plant (VPP) is proposed by using Smart Contracts on Ethereum Blockchain Platform. The P2P energy trading is the recent trend the power society is keen to adopt carrying out several trial projects as it eases to generate and share the renewable energy sources in a distributed manner inside local community. Blockchain and smart contracts are the up-and-coming phenomena in the scene of the information technology used to be considered as the cutting-edge research topics in power systems. Earlier works on P2P energy trading including and excluding blockchain technology were focused mainly on the optimization algorithm, Information and Communication Technology, and Internet of Things. Therefore, the financial aspects of P2P trading in a VPP framework is focused and in that regard a P2P energy trading mechanism and bidding platform are developed. The proposed scheme is based on public blockchain network and auction is operated by smart contract addressing both cost and security concerns. The smart contract implementation and execution in a VPP framework including bidding, withdrawal, and control modules developments are the salient feature of this work. The proposed architecture is validated using realistic data with the Ethereum Virtual Machine (EVM) environment of Ropsten Test Network.

INDEX TERMS Bidding system, blockchain, Ethereum, peer-to-peer (P2P) energy trading, smart contract, virtual power plant (VPP).

NOMENCLATURE

ABI	Application Binary Interface
AET	Average Execution Time
dApp	Decentralized Application
DER	Distributed Energy Resources
DSO	Distribution System Operator
ESS	Energy Storage System
ETC	Energy Trading Coordinator
ETH	Ether, Ethereum
EV	Electric Vehicle
EVM	Ethereum Virtual Machine
ICT	Information and Communication Technology
IDE	Integrated Development Environment
P2P	Peer-to-peer
PoS	Proof-of-Stake
PoW	Proof-of-Work
PV	Photovoltaic
RES	Renewable Energy Sources
RS	Running Scheme
$t_{tx_{confir}}$	Transaction Confirmation Time
$t_{tx_{input}}$	Transaction Request Time
TG	Total Generation
TL	Total Load
TSO	Transmission System Operator
VM	Virtual Machine
VPP	Virtual Power Plant
Web3.js	Ethereum JavaScript API

The associate editor coordinating the review of this manuscript and approving it for publication was Yanbo Chen.

I. INTRODUCTION

Distributed generation is electricity production from variety of distributed energy resources (DER) such as rooftop solar photovoltaic (PV) units, wind generating units, open and closed cycle gas turbines, diesel generators, hydro or mini-hydro schemes, and battery storage. In contrast to the conventional electric power systems, distributed generation is alterable, amenable, acentric, and customizable owing to its structure adjacent to the ultimate consumer spot. DER are mostly arranged in microgrids which are being either connected or disconnected from grid [1].

Microgrids consist of localized set of power sources, loads, and DER. In few last decades, the notion of microgrids had been becoming more common and many microgrids operated in such a way to use energy effectively and efficiently. Some studies have been conducted to integrate DER into grid while guaranteeing system operations satisfactorily [2], [3]. The Virtual Power Plant (VPP) concept has been raised afterward to be able to incorporate DER into the grid enabling bi-directional power and information exchanges without affecting grid reliability and stability, utilizing the blessings of Information and Communication Technology (ICT) [4]. It is theoretically used for aggregation of DER, so that they can serve as a fully dispatchable unit managing information from a wide variety of physical infrastructures such as wind, hydro, solar photovoltaics (PVs), Energy Storage Systems (ESSs), market operation, and distribution system operator (DSO).

Majority of the electricity customers, known as consumers in microgrid, VPP, and power system are connected with typical centralized energy trading systems where the energy trading is handled in wholesale markets regulated by the transmission system operator (TSO). On the other hand, the modern power systems including microgrid and VPP accommodate many DERs where the concept of energy consumers has been changed as prosumers, who can conceptually produce and consume energy. The generation of electrical energy by the components of DERs is stochastic and intermittent; therefore, the prosumers of the VPP who have surplus energy can store it if they have energy storage systems, or sell it to the grid or other parties. This transaction of energy among prosumers is called Peer-to-Peer (P2P) energy trading [5]. In the smart grid framework, the energy trading algorithms are becoming important factors to fulfill the energy demand requirements considering the unpredictable generation pattern of DERs. These days, game theory has been identified as a potential analytical tool for energy trading and sharing in microgrid and smartgrid which mathematically allows solving optimization problems with multi-objective functions [6]–[11]. A new scalable market design for P2P energy trading through bilateral contract networks is reported in [12]. In [13], an incentive prosumer based P2P energy trading is proposed.

It appears that P2P energy trading in VPP framework is relatively new and most of the reported works are either in the conventional ICT domain or using optimization algorithm to handle unpredictability of microgrid operations. Only few works and real-world projects in microgrid and VPP domain are focused on decentralized mechanism using public blockchain technology.

In this work, a novel VPP architecture has been developed to enable P2P energy trading mechanism with auction-based bidding model using smart contracts and priority was given to explain the stages of development and implementation over Public Ethereum Platform. Unlike others, this platform can be used among other VPPs and intra-VPPs since public blockchain is used and it is relatively scalable and less costly compared to ICT operations needed for private blockchain usage for every single VPP. Because there is no need for keeping in-house servers and nodes up to create a private blockchain network. Essentially, the need for intermediary authorities such as aggregators in the use of private blockchain undermines the true decentralization and transparency concept of P2P trading. To reach that adaptivity, in this article, public blockchain environment is chosen over private and consortium (permissioned) blockchain networks because of high initial cost and limitations on physical structure, respectively.

The proposed platform is implemented based on the needs of the VPP framework and includes several modifiable mechanisms for easy adaptation to the different inter- or intra-operations of VPP operators. In the auction mechanism, bidding, withdrawal, and control modules are developed to show the operability of the platform. Three different running schemes (RS) are considered and proposed also to address centralization, cost and security measures in P2P energy trading. A complete smart contract platform and real-life cryptographic testing environment have been realized using EVM, Remix, Metamask, Web3.js, Infura.io, Ropsten and the P2P energy trading in VPP is verified using realistic generation and load data. The contributions of the paper can be summarized as follows:

- The proposed solution of P2P energy trading is solely for VPP architectures and new in VPP domain.
- The implementation part is demonstrated step by step by integrating power system and blockchain ecosystem, as well as presenting several running schemes.
- A modular smart contract mechanism is proposed which can be used effectively in P2P energy trading within VPP framework. Thus, each module can be improved and be easily adapted to several other use cases.
- Usage of public Ethereum network instead of private or consortium network and modular approach to VPP framework to make it applicable to both intra and inter VPPs; systematic way to implement smart contract to enable P2P trading, development of bidding, withdrawal, and control modules by properly integrating power system and computing software are the novel contributions of this article.
- Therefore, unlike other studies, the proposed framework and approach in this article is able to be adapted and converted easily to Decentralized Application (dApp) which is the cutting-edge usage of smart contracts and blockchain. DApps are expected to be an important part

of the new era in the world-web history, which is named as Web 3.0.

II. MODERN ENERGY TRADING APPROACHES AND BACKGROUND

Currently, there are quite a few projects and initiatives enabling trading between consumers and prosumers possible in microgrids by the help of the conventional ICT, mostly using client-server architecture [14]. The most of the energy management and trading platforms had been created by using these technologies are aiming general wholesale or retail business models [15]. Correspondingly Porto and US based two initiatives with the same name as Smartwatt, UK based Piclo, Netherlands based Vandebron and German project Smart Watts can be given as examples and these systems attempt to reach economic efficiency by making the trading easy and optimized [16]–[19]. Furthermore, Sonnen community [20] set their goal as sharing and trading energy in order to fulfil energy needs from RES in a decentralized manner. Therefore, it is easy for one to interfere that the inclination in the power market is towards P2P sharing and trading. Because eliminating the intermediaries brings efficiency to the grids in terms of time, cost, and effort spent.

Potential instances of P2P usage include decentralized trading, i.e., mutual trading among prosumers, consumers, and conventional power suppliers. Hence, PeerEnergyCloud project's objective in Germany was making research and development of cloud-based technologies for such a concept [21]. This covers the creation and the implementation of an advanced recording and prediction methodology to address the local excessive energy production issue, including a virtual marketplace for local energy trading.

While above-mentioned industrial projects are mostly the traditional server-centric and have a central authority to control the operation, energy trading efforts migrate towards blockchain since the intrinsic decentralizing nature of the blockchain architecture is consistent with the decentralized P2P trading. As a matter of fact, London-based energy technology company Electron [22] developed an energy metering and billing platform using blockchain. And TransActiveGrid, which afterwards took place under the umbrella of US-based energy technology start-up LO3 [23], established Brooklyn Microgrid successfully as the first P2P energy trading project within microgrids by using blockchain [24]. Blockchain-based P2P energy trading companies akin to Power Ledger were established and projects similar to White Gum Valley project were realized in Australia [25]. The country has great potential for decentralized P2P trading thanks to its solar insolation, wind power sources and its relatively high-cost grid-sourced electricity [26].

There are several power-based applications that leverage blockchain platforms including data exchange scenarios between smart devices, digital P2P transactions, machine-to-machine (M2M) communication, business-to-business (B2B) energy trading, mutual transactions between prosumers and consumers in *transactive energy networks*, smart home, electric vehicle (EV), and microgrid development scenarios [27].

The usage of blockchain can contribute to fulfilling the strict security and privacy requirements of the IoT systems for local electricity storage systems. Hence, significant research studies focused on anonymous payment and safeguarding peers or EV owners' privacy on the trading platform. Kang *et al.* [28] have come up with a localized P2P electricity trading system with consortium (permissioned) blockchain. Trading among plug-in hybrid electric vehicles (PHEVs) in smart grid is realized with an iterative double auction mechanism. In [29], a security model for trading between EVs and charging pile management on the blockchain that leverages the lightning network and smart contract technologies was focused. A decentralized energy trading system with blockchain was presented using multi-signatures to enable peers to perform transaction anonymously and securely [30]. In [31], a credit-based payment scheme and a Stackelberg game based optimal pricing strategy were proposed to support the scalability of transactions. A consortium blockchain is used for the security concerns. A local energy market operated with a double auction system that uses a smart contract on a private Ethereum blockchain to determine the market closing hours have been developed [32]. Nonetheless, limited information regarding the implementation of the smart contract and how the price is cleared during each trading session was given. An energy-trading system has been developed using consortium blockchain so that it could be secure and privacy-preserving in the smart grid [33]. In [34], a blockchain based P2P energy trading and crowdsourcing architecture with an optimization model is developed. In [35], all transactions are stored on a consortium blockchain which is generally supervised by some kind of aggregators or energy traders and the financial institutions that support anonymous payment. In [36], P2P transactions between EVs and grid, and among EVs realized with an EV power trading model based on private Ethereum blockchain and smart contract, considering the randomness and uncertainty of the EV charging and discharging. A reverse auction mechanism based on a dynamic pricing strategy and aggregators is used. In [11], a P2P energy trading scheme with the cooperative Stackelberg game formulation was proposed to help a centralized power system to reduce the total electricity demand at the peak hour. Price-based control of DERs to support the grid is also a matter of concern in VPP-related literature. Di Silvestre *et al.* [37] studied ancillary services in the energy blockchain for microgrids and focused mainly on the technical issues related to power transmission. In [38], again it's focused on secure and verifiable energy trading with blockchain. In the study, it's emphasized that the blockchain should provide transparency, immutability, and auditability to the energy trading. A consortium blockchain based scheme was proposed to block energy sellers refusing to transfer the negotiated energy to the purchaser. In [39], a consortium blockchain is used to design a hybrid P2P energy trading market where consumers and prosumers trade each other and with the main grid. Although the study clearly elaborates on the concepts of P2P trading in a smart grid environment, it lacks the implementation details regarding blockchain and smart contracts and shows simulation results with local machine development tools of Ethereum. Han *et al.* [40] proposed a private Ethereum based smart contract architecture with the conventional double auction. A smart contract consisting of four core algorithms has been developed; the purpose of each algorithm is to save gas consumption and ensure security. Performance measures are given, energy trading supports 25 agents at the same time, with more than six miner nodes. In [41], introduces a private Ethereum blockchain based energy trading architecture for EVs within smart cities. It is not a clean slate approach and builds on to the existing infrastructure. Although transparency brought by blockchain is praised, the private version is used because it is considered to be more efficient than the public version. Also, it is noted that executing a large number of transactions causes a serious computational load.

In some scenarios, security can be the bottleneck due to the variety of the participants, however, in some cases, the architecture and transparency could be the key point for the platform. Please note that, unlike the trading environment of EVs, in a VPP environment agents are mostly stationary and naturally, there is a limitation for participating in the blockchain network because of the physical requirements i.e., power lines and infrastructure. Also, all the participants and roles are needed to be known by the VPP admin, which eliminates the random users, to assure connectivity and reliability for distribution network. Therefore even if the public blockchain is used for the trading system, it is being restricted by the physical conditions, and benefiting the advantages of the public network simultaneously.

According to the survey published by German Energy Agency, in power market and electricity value chain establishing smart contracts can be utilized for demand response services, cooperation and control of VPPs, grid and network, governance of energy storage systems, control of decentralized energy systems, community energy projects, and coordination of RES power plant portfolio [42]. There are some concerns and costs regarding the adaptation of the current power infrastructure to work with blockchain and smart contracts, i.e., deploying compatible smart meters and Internet of Things (IoT) appliances. However, the business processes for energy trading can likely be reconstructed by this trend, together with the capability of automation and big data analytics. Using this information analysis could yield to demand aggregation and response services being optimized, could promote VPPs, and possibly improve the involvement of active consumers, prosumers, and renewable energy.

This study is focused to resolve the business processes associated with P2P energy trading of VPP.

III. BLOCKCHAIN AND SMART CONTRACT

In the last decade, when the peer-to-peer (P2P) money transaction is introduced in [43] without any intermediary authority such as banks, many cryptocurrencies mushroomed. The technology behind the cryptocurrencies, known as the Blockchain, leads many other future promising applications as well [44]. Blockchain is a distributed platform with interconnected blocks which constitute a vast immutable digital ledger in the end. The integrity and consistency of transactions are protected by cryptographic mechanisms such as hash functions, asymmetric encryption (public-key cryptography), and Merkel-trees [45]. All the transactions are kept on the blocks just like the traditional bank records with the difference of generating a distributed universal public ledger *eventually*. Every block, except the first one known as genesis block, points out the previous block with its hash in order to create a chain of blocks. The entire system is based on a P2P network. Nodes keep the database distributed and decide which transactions will be approved. Since they work for the liveliness of the system, participants get rewards, which is called mining. Therefore, the blockchain becomes a very distinctive kind of immutable distributed large-scale database and used in several fields in addition to finance. Ethereum is one of these blockchain-based platforms and differentiates itself by being capable of running programmable transactions, i.e., smart contracts on the system [46].

A. CONSENSUS ALGORITHMS

Distributed consensus algorithms are used to keep the truly decentralized structure of the network.

The certain algorithm that is used to reach consensus among the network nodes, affects key parametric of that blockchain network such as scalability, transaction speed, security and even electricity consumption of the nodes. There are trade-offs between their certain advantage and disadvantages. Although there are variety of consensus algorithms, either of Proof-of-Work (PoW), Proof-of-Stake (PoS) or modified version of these two are generally in use of the majority of the blockchain applications. In general, every algorithm needs a way to generate blocks and accept the proposed block by network members, a process called *reaching consensus*. Using a PoW-based blockchain network, e.g., Bitcoin, is not very suitable, especially for energy applications, because of the computational power and energy consumption. On the other hand, Ethereum uses a hybrid version of PoW and PoS. However, it's stated that the Ethereum platform is planning to use PoS or slightly modified version of it with the version of ETH 2.0 in a couple of years to reduce the energy and resource consumption [47].

B. ETHEREUM AND VIRTUAL MACHINE

Ethereum is an open-source project developed by many people around the world and not controlled or owned by any particular person. It is not solely for storing or transferring value as its most counterparts. The main aim is to make anyone capable of building or using decentralized applications that run on blockchain technology [47]. Ethereum Virtual Machine (EVM) is in the center of the platform as a runtime environment, as it's fundamentally a level of abstraction between the machine and executing code. EVM helps development and portability of the code because it is an exquisite sandbox, a testing environment for those trying to create a smart contract without affecting the main blockchain operations. The remainder of the main network is fully isolated from an EVM instance. In the network, any Ethereum node can execute the same commands on their own EVM that provides code portability.

Speaking of resources, in EVM, there is a fee named ‘gas’ for computational cost of running certain piece of opcode on the network in order to prevent the denial of services attacks and increase the efficiency of the system.

Users can participate in the public Ethereum network and pay ‘gas’ to miner nodes, or a new private network can be created with permissioned miner/user nodes. In a public network, there is transparency, and the performance of the system is depended on the execution of the global network. On the other hand, in a private network, there is an initial ICT cost for servers and network, and a maintenance cost as well to have enough miner nodes, and it can cause centrality to certain degrees when more substantial nodes take the lead. In these networks, there are several factors, i.e., consensus algorithm, delay, number of nodes needed to be measured to show the performance. There are also semi-structured Ethereum networks, e.g., consortium blockchain that binds public and private networks on the same platform. There are pros/cons for public and private blockchain networks where the consortium blockchain is placed in the middle of these two architectures. To run consortium blockchain, there is a group of privileged nodes takes the lead over other participants.

C. UNDERSTANDING SMART CONTRACT

Smart Contract concept was envisioned by Nick Szabo as a computer-aided set of rules that provides an agreement in a group of peers. Vending machines are illustrated as a forefather of the smart contracts as it is a “contract with a bearer” [48]. Smart contracts today are able to work autonomously on the Ethereum-based blockchain platforms that allow executing immutable digital agreements. In these platforms, agreement protocol among the contractors is initially implemented with a script and deployed to the network. When a specific data or command occurs, the deployed smart contract is being triggered automatically on the blockchain network, and the actions in this digital contract is followed. Thus, the whole business is completed transparently without needing trusted central authorities.

Smart contracts can be considered as wallets in the cryptocurrency concept, since they have an account address and balance akin to standard cryptocurrency accounts. Hence, all other participants can transfer value between their own accounts and the smart contract. The only difference is that the business workflow, the protocol between the parties, is programmatically coded inside the smart contract. A function call or a transaction triggers the smart contract execution if the business logic holds at that time. When a new smart contract is implemented, it must be deployed to the Ethereum network. This process and all other execution steps are done by the peer nodes in decentralized concept. Thus, while deploying a new contract, or executing one, the system charges a little fee to handle these processes, which is called gas.

Creating new applications on the Ethereum platform is relatively easy and suitable for many real-world scenarios. Smart contracts are robust to interventions from outside since they are deployed to a blockchain, and kept in blocks anonymously, yet all the transactions can be monitored and traced publicly. Smart contracts have a value (essentially, it is a balance in Ethereum), an address, state, and functions that can change the state during the operation and eventually emits output events. These events can be captured by the external web or mobile applications so that the *dApps* come to life. It is highly likely that *dApps* will embody the *Web 3.0* infrastructure in the near future [46].

IV. PROPOSED ARCHITECTURE OF BLOCKCHAIN BASED P2P ENERGY TRADING IN VPP

The VPP architecture requires known participants and power lines during the trading process to guarantee power distribution among all known users. Thus, the blockchain based solutions can not be truly decentralized because of the oracle problem, and it naturally has some limitations on participation. Instead of using consortium blockchain platforms, e.g., Hyperledger, Quorum, or modified private Ethereum network, the public Ethereum network is being set to make the whole process transparent and adaptable to various backbone architectures. Hence, the platform can run communication and power distribution processes on different rules or networks efficiently. In this work, the communication and agreement are moved on to the public Ethereum network, which gives the adaptability. Public network transparently decides a pair of participants to assure power transfer between them, and it occurs when users become a part of the physical network. With this structure, it is also possible to run inter-VPP energy trading while we are offering an example of intra-VPP distribution in this work.

Fig. 1 shows the proposed VPP model consisting of twelve agents, including consumer/prosumer, a big scale Energy

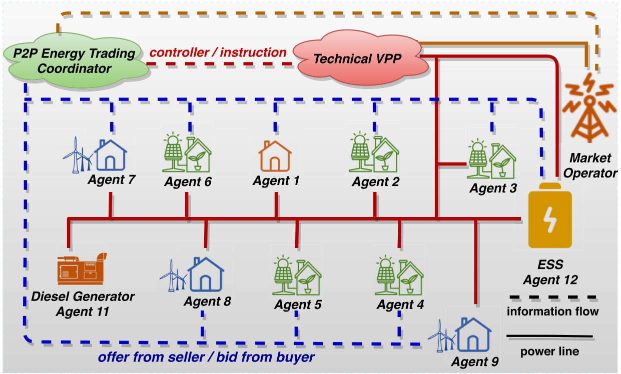


FIGURE 1. VPP model architecture.

Storage System (ESS), a Diesel generator, and a P2P Energy Trading Coordinator (P2P_ETC). The VPP is controlled by P2P_ETC and technical VPP, and it is also connected to the upper level entity, named as Market Operator to enable P2P energy trading. P2P_ETC is responsible for financial issues, e.g., investment, optimized revenue for exchanged energy, economical paradigms with ancillary grid services and participates in auction mechanism. It relies on agents' information shown with dashed line in Fig. 1. Technical VPP (TVPP), as the name suggests, handles technical issues relevant to controls at agent and VPP level. There are information and power flow between the agents as shown in dashed and solid line in Fig. 1. In order to make energy trading efficient, a bidding system between the agents that runs on the blockchain network is built. The Ethereum platform and smart contracts for these purposes are utilized. Every agent has public and private key pair to have an address on the platform.

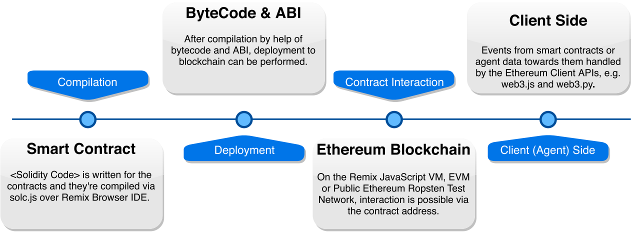
In this proposed architecture, agents mostly have a prosumer role since they are able to produce energy from renewable sources, i.e., solar, wind when it is available. On the other hand, their role might be changing to a consumer in parallel when it is not available from the grid in accordance with VPP operation. When the agents have a surplus of energy, they will be able to sell those to other needful agents in VPP or connected VPPs. The seller agent will initiate the auction by deploying a smart contract, and buyers will bid for it to get the energy they require.

A. SMART CONTRACT IMPLEMENTATION

Two smart contracts allow the system to handle a bidding mechanism between agents. The smart contracts are developed with the Solidity programming language and Remix browser IDE (integrated development environment). The stages of development and testing are summarized in Fig. 2. The smart contracts are compiled by using solc.js over Remix Browser IDE [49]. By doing so, the bytecode and ABI (Application Binary Interface) of the smart contracts are generated. After this step, the bytecode can be deployed to the public or private blockchain test environments or real-time environment. A testing environment of Remix Browser's JavaScript VM, and Ethereum Ropsten Test Network is used [50].

Implemented *auction contract* has a straightforward interface, allowing agents to place bids and withdraw funds after the auction ends. In unexpected situations, the auction owner must be entitled to cancel the auction and to withdraw the winning bid. There has to be an *auction owner* to whom the winning bid will go when the auction finishes successfully. The auctions must have a start and end time. The block numbers can adjust this period since it is not safe to use block timestamps, which are set by miners and can be easily spoofed. Ethereum blocks are generated in approximately every 15 seconds, so the duration of auction can be calculated from these estimates instead of the easily modifiable timestamp fields of blocks.

In auctions, users try to bid the maximum amount that passes the highest bidder of the auction. Although there exist many different approaches for realizing the auctions in the literature, ‘open English Auction’ workflow is adopted in this study, giving the focus on smart contract implementation hurdles behind the approach to enable a successful P2P system within VPP framework [51], [52]. With the usage of smart contracts, it is needed to reduce the gas price for economic operation as well as increase the security, privacy, and transparency at the same time. According to the needs of VPP, this scheme seems the fair solution since other complex mechanisms can cause costly operations and code-security breaches of Solidity during Smart Contract implementation. Writing a smart contract is straightforward in terms of programming. On the other hand, avoiding logical, operational, and financial flaws in smart contracts that have complex mechanisms are difficult and significant. Due to this, “auditing smart contracts” is becoming another special job description and requirement while developing distributed applications. Nevertheless, the proposed platform can be applied with different auction mechanisms with a small update as a modular design approach is followed in this study. Following are the essential key elements of the adopted auction:



After compilation by help of bytecode and ABI, deployment to blockchain can be performed.

Events from smart contracts or agent data towards them handled by the Ethereum Client APIs, e.g. web3.js and web3.py.

<Solidity Code> is written for the contracts and they're compiled via solc.js over Remix Browser IDE.

On the Remix JavaScript VM, EVM or Public Ethereum Ropsten Test Network, interaction is possible via the contract address.

FIGURE 2. Overview of proposed smart contract development and testing platform.

- Increment: The bid increment amount which is set by the auction owner in the beginning.
- HighestBidLevel: Current highest bidding level, which will be the amount to pay when the auction finishes for the highest bidder.
- HighestBid: The highest bid that so far has been put in the auction.
- HighestBidder: The agent who made the highest bid until the current time.

When a new bid is greater than the previously highest one, the current highest bidding level is calculated as the previous top added to the bid increment amount. With this algorithm, the fairness of the competition is secured; otherwise, rich participating parties could overact easily to win all the auctions. Algorithm 1 summarizes the whole pipeline clearly.

The auction smart contract works on top of four main modules. The user roles and implementation details for the public procedures are given as follows:

1) INITIALIZATION/CONSTRUCTION MODULE

This module controls certain preconditions, then sets some variables in the storage of the contract. For instance, during the creation of a new auction, the start time and end time must

Algorithm 1 Part of Auction Algorithm

```
Require: $highestBid \geq newBid > 0$
$newAmount \leftarrow newBid + increment$
if $newBid \leq highestBid$ then
    $highestBidLevel \leftarrow \mathrm{MIN}(newAmount, highestBid)$
else
    if $msg.sender \neq highestBidder$ then
        $highestBidder \leftarrow msg.sender$
        $newHigh \leftarrow highestBid + increment$
        $highestBidLevel \leftarrow \mathrm{MIN}(newBid, newHigh)$
    end if
    $highestBid \leftarrow newBid$
end if
```

be proper. The start time must be before the end time, and end block number must be bigger than the current block number. Whenever agents want to initialize an auction, they have to deploy the auction contract with constructor parameters. Auctions must have an *owner* for each deployed contract; otherwise, it would not be possible to withdraw the funds. According to the running schemes, the P2P_ETC can be the only agent to have the ability to start a new bidding period, or each agent can deploy their own auction with their parameters.

2) BIDDING MODULE

Making a new bid is not acceptable before the starting time, and after ending time or when the auction is canceled. It is very critical to block off the auction owner from making bids to their auctions. The owner can increase the price and manipulate the bidding to earn more. When a new auction starts, any agent can attend the bidding if there is no restriction rule made by the P2P_ETC. In Solidity programming language, the programmer can impose these restrictions by using reusable function modifiers. Making modifiers as simple and straightforward as possible they can be, helps to use them together in an efficient way. Users are able to send ETH (Ethereum) to make bids with this function. There may be cases that bidders need to withdraw their ETH:

- if someone makes a bid more than the highest bid.
- if someone makes a bid more than the highest bid level but less than the highest bid.

The smart contract does not automatically refund the funds; instead the withdrawal module is used due to the security considerations. The smart contract sends ETH to a user when they explicitly request a withdrawal after all this bidding period is end [53].

3) WITHDRAWAL MODULE

Upon completion of an auction, canceled or not, bidders should hold the ability to take their money (ETH) back. Only the auction participants can use this module for a withdrawing process. The cases that have to be handled by the smart contract for the requests are given as follows:

- the owner who opened the auction should be able to withdraw the ETH amount of the highest bid level since that is the award of the winner.
- the highest bidder should be able to withdraw their excessive part, which is the maximum bid minus highest bid level.
- excluding these two cases and users, any amount of ETH that sent to the smart contract should be able to withdrawn.

4) CONTROL MODULE

When the system detects any fraudulent activities from the agents or the system itself, somehow, it cancels the whole pipeline automatically. It is implemented with the help of assorted modifiers in a smart contract. For instance, a canceled flag is changed to true under certain conditions such as *'only before end'* and *'by only owner'*.

In contrast to other programming contexts, writing Solidity contracts usually necessitates fewer lines of code, but attention to a great deal of detail. Until there are better tools to analyze security, gas, and readability considerations, which are very vital, developers will carry entire burden on their shoulders.

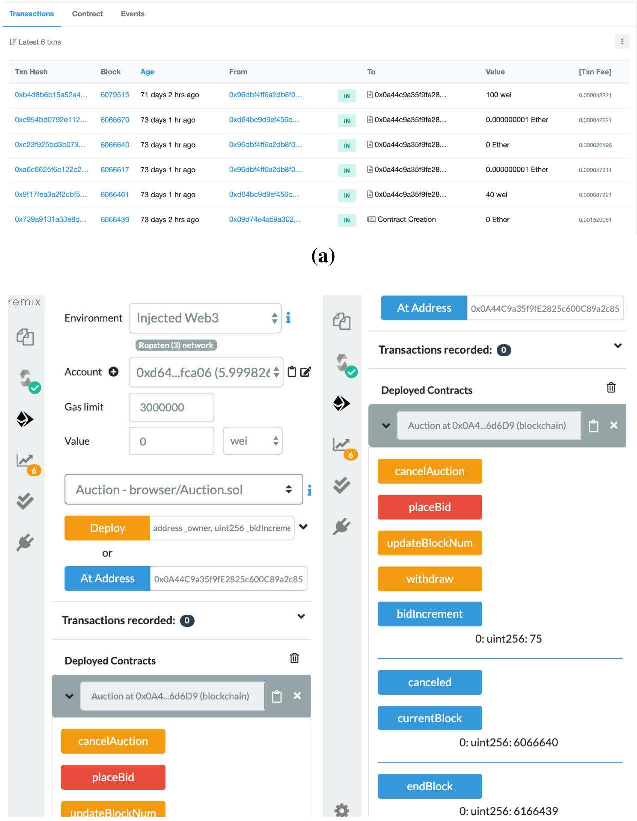
B. EXECUTION

Once the smart contracts have been implemented, to use or invoke them, they should have been deployed into the Ethereum platform. In our proposition, they are being developed, tested, and deployed on to the JavaScript EVM of the Remix. The ABI or bytecode of the contract can be obtained from the compilation plug-in part of the Remix IDE. Afterward, in order to reach a real-time simulation of the implementation, they have been deployed to the Ropsten, which is a public Ethereum Blockchain test network.

In Fig. 3, a sample execution of the contract is shown. First, three Ethereum test accounts had been created in Metamask, and their balance filled from some faucets. Faucets are third-party websites that are used to get some ethers (ETH) directly to related test network account address for testing purposes. Then the contract is deployed to Ropsten by using Metamask, which is a Web3 injection extension for the browsers and Remix. For contract creation *1520051 gas unit* was used, and the gas price is in gwei (1 / 10^9 Ether), so it makes 0.001520051 Ether for deployment cost which converts to \$0.32 as of September 2019. Let us assume, Account1 is the owner of the contract and deployed it with the arguments as, bidIncrement: 75, startBlock: 1 and endBlock: 100000. In respective order, Account2 bids 40 wei (1 / 10^{18} Ether), Account3 bids 1 gwei, uses a not payable method without sending any value. In respective order, Account2 bids 40 wei (1 / 10^{18} Ether), Account3 bids 1 gwei, uses a not payable method without sending any value. As it is summarized in Table 1, this time Account2 bids 1 gwei, and its total bid becomes 1gwei + 40 wei. After that, Account3 bids 100 wei to win the auction. In the end, the highest bid becomes 1000000100 wei, Account3 becomes

TABLE 1. Transactions and gas costs.

From	To	Event	Fee (Gwei)
0x09d74e4a59a302...	0x0a44c9a35f9fe28...	Account1 deployed the contract, bidIncrement:75	0.001520051
0xd64bc9d9ef456c...	0x0a44c9a35f9fe28...	Account2 bids 0.00000004 Gwei	0.000087221
0x96dbf4ff6a2db8f...	0x0a44c9a35f9fe28...	Account3 bids 1 Gwei	0.000057211
0xd64bc9d9ef456c...	0x0a44c9a35f9fe28...	Account2 bids 1 Gwei and its total bids and highest bid become 1.000000040 Gwei	0.000042221
0x96dbf4ff6a2db8f...	0x0a44c9a35f9fe28...	Account3 bids 0.0000001 Gwei, becomes highest bidder and winner	0.000042221



Latest 6 txns

Txn Hash	Block	Age	From		To	Value	[Txn Fee]
0xb4d8b6b15a52a4...	6079515	71 days 2 hrs ago	0x96dbf4ff6a2db8f0...	IN	0x0a44c9a35f9fe28...	100 wei	0.000042221
0xc954bd0792e112...	6066670	73 days 1 hr ago	0xd64bc9d9ef456c...	IN	0x0a44c9a35f9fe28...	0.000000001 Ether	0.000042221
0xc23f925bd3b073...	6066640	73 days 1 hr ago	0x96dbf4ff6a2db8f0...	IN	0x0a44c9a35f9fe28...	0 Ether	0.000026496
0xa6c6625f6c122c2...	6066617	73 days 1 hr ago	0x96dbf4ff6a2db8f0...	IN	0x0a44c9a35f9fe28...	0.000000001 Ether	0.000057211
0x9f17fea3a2f2cbf5...	6066461	73 days 1 hr ago	0xd64bc9d9ef456c...	IN	0x0a44c9a35f9fe28...	40 wei	0.000087221
0x739a9131a33e8d...	6066439	73 days 2 hrs ago	0x09d74e4a59a302...	IN	Contract Creation	0 Ether	0.001520051

(a)

Environment: Injected Web3
Ropsten (3) network
Account: 0xd64...fca06 (5.99982€
Gas limit: 3000000
Value: 0 wei
Auction - browser/Auction.sol
Deploy: address _owner, uint256 _bidIncreme
or
At Address: 0x0A44C9a35f9fE2825c600C89a2c85
Transactions recorded: 0
Deployed Contracts
Auction at 0x0A4...6d6D9 (blockchain)

(b)

At Address: 0x0A44C9a35f9fE2825c600C89a2c85
Transactions recorded: 0
Deployed Contracts
Auction at 0x0A4...6d6D9 (blockchain)

0: uint256: 75

0: uint256: 6066640

0: uint256: 6166439

(c)

FIGURE 3. Proposed smart contract in Remix. (a) executed transactions in Ropsten Test Network, (b) deployment, and (c) running.

the highest bidder, and the smart contracts balance becomes 0.00000000200000014 Ether. Since a big interval for start and end blocks was set, the auction continues for a quite long time. It should be remembered that the accounts (agents) has to withdraw their related balances from the smart contract once the auction ends or canceled by the owner.

In a public blockchain network environment, it is possible to verify and publish the contract source code. Verification of source code and uploading it to the system gives extra transparency for all. Like normal agreements, a smart contract should provide more data to both parties regarding what they are *digitally opting* for and offer them the chance to audit the code independently to confirm that they are genuinely doing what they are meant to do.

C. RUNNING SCHEMES

The proposed smart contract-based bidding platform can be adapted to different running schemes. In the given general scenario shown in Fig. 4, once an agent has excessive energy to offer, it will inform the CVVP. After that, there could be three approaches to start a new auction:

1) CENTRALIZED APPROACH - P2P_ETC DEPLOYS

P2P_ETC itself deploys the smart contract periodically for definite durations for buying or selling windows. This approach may cause the centralization problem, which contradicts the blockchain and P2P phenomena. P2P_ETC checks its database and energy profile in order to decide when the RES generate more energy, and there is excessive energy available for P2P trading within the VPP. Accordingly, P2P_ETC starts the auction by deploying the smart contract periodically for each auction. For example, around noon, when there is enough daylight to generate energy, P2P_ETC can have 20 minute-long buying auctions for every hour to collect energy from the producers. After that, in the rush hour, P2P_ETC starts selling auctions for the consumers, again with 20 minute-long periods.

2) SECURE APPROACH - AGENT DEPLOYS

Agent itself deploys the smart contract, which could be safer but costly due to the initialization process. In this scenario, an agent, e.g., Agent 4, checks its smart meter and the system. Once the agent has excessive energy to sell others, informs P2P_ETC for starting a new auction. Using its database and current energy profile does P2P_ETC make the decision to let the agent start this bidding period. When P2P_ETC approves the request, the agent deploys the smart contract using its bytecode and ABI, which is already given the agents. The auxiliary software or operators of agents interact with smart contracts, i.e., deployment, setting parameters, or input of preferences with Ethereum Client APIs, e.g., *web3.js, web3.py*. Transactions towards Ethereum platform and events from there, are transferred over the network via HTTP (Hyper Transfer Protocol). When the auction is deployed, other agents who want to join the auction can make a bid to join the process in certain conditions, which is explained in section IV-A.

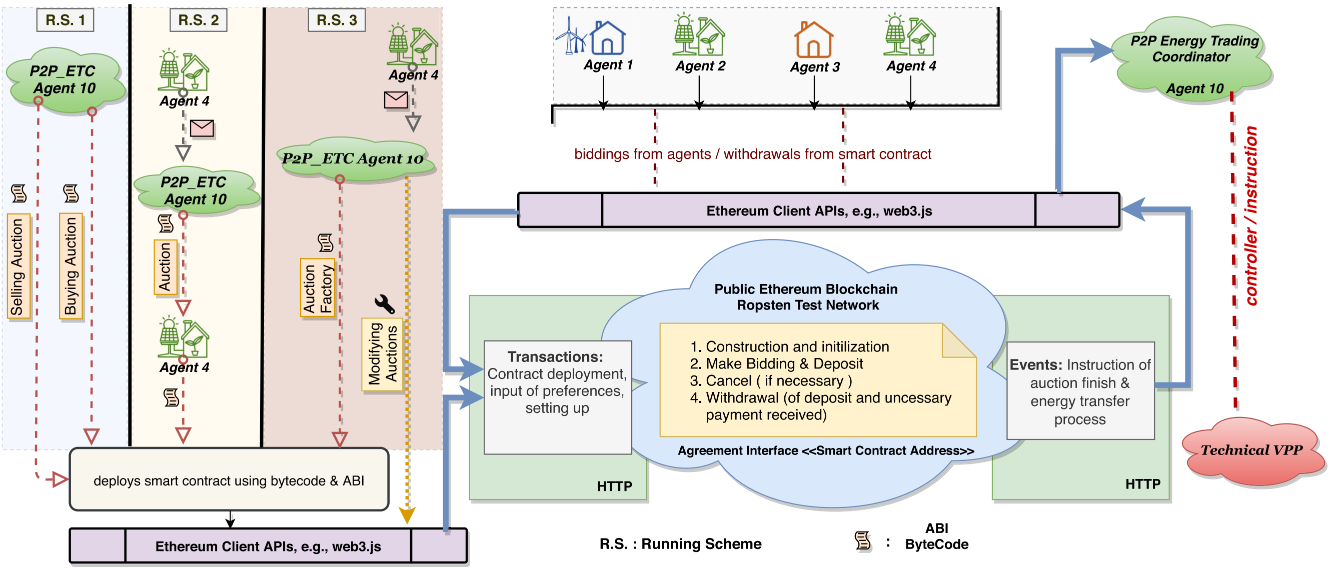

R.S. 1
R.S. 2
R.S. 3
P2P_ETC Agent 10
Selling Auction
Buying Auction
Agent 4
P2P_ETC Agent 10
Auction
Agent 4
Agent 4
P2P_ETC Agent 10
Auction Factory
Modifying Auctions
deploys smart contract using bytecode & ABI
Ethereum Client APIs, e.g., web3.js
Agent 1
Agent 2
Agent 3
Agent 4
biddings from agents / withdrawals from smart contract
Ethereum Client APIs, e.g., web3.js
P2P Energy Trading Coordinator
Agent 10
controller / instruction
Public Ethereum Blockchain
Ropsten Test Network
Transactions: Contract deployment, input of preferences, setting up
1. Construction and initilization
2. Make Bidding & Deposit
3. Cancel (if necessary)
4. Withdrawal (of deposit and unecessary payment received)
Agreement Interface <<Smart Contract Address>>
Events: Instruction of auction finish & energy transfer process
HTTP
HTTP
Technical VPP
R.S. : Running Scheme
ABI ByteCode

FIGURE 4. Running schemes, flow and model architecture used in the proposed P2P energy trading.

3) ECONOMY APPROACH - P2P_ETC ADJUSTS PARAMETERS

P2P_ETC deploys the smart contract factory that generates smart contracts on behalf of the agents. Here the smart contract named *Auction Factory* is used to create auction smart contracts to reduce initial costs. When a new auction request is submitted, P2P_ETC will adjust the parameters for the smart contract which is already deployed. With this approach, it is possible to avoid the deployment gas cost, but it may cause security risks due to the transparent background of the blockchain network. The whole communication among the agents and the P2P_ETC is made over the system via HTTP (Hyper Transfer Protocol). The proposed platform can run all these running schemes. P2P_ETC will decide to operate one of these schemes, according to worthwhileness to the system overall, i.e., cost or security.

D. SECURITY DISCUSSION

In some other peer to peer energy trading applications, like vehicle-to-vehicle (V2V) or vehicle-to-grid (V2G) networks, privacy is very crucial since previously unknown EVs can come to charging stations to participate in the auctions [28], [36], [54]. At this point, the privacy brought by a private blockchain and sealing bids in auctions becomes important as well. However, in a VPP environment, there is no need for strict privacy or security measures like those platforms that have random participants. In the proposed framework participating agents are already known by the P2P_ETC and TVPP because of the backbone architecture. TVPP and P2P_ETC are in charge of all power transactions and financial transfer operations. Please note that, if the system has a potential random participant, in that case, the TVPP can not assure to transfer energy to those nodes since they are not a part of the physical network. Bids made by the agents, cannot be tracked easily in a huge, real decentralized public Ethereum network in which also Ether cryptocurrency transfers are conducted. Ether is also widely used outside of the energy sector, therefore agents can use it outside of P2P trading and directly reimburse, and unlike permissioned or private blockchain, TVPP and P2P_ETC will not be able to misbehave the tender because they will not be authorities that have privileges on the blockchain network [55]. As can be seen very clearly from the current literature, there is a trade-off between public blockchain and private blockchain usage. Namely, privacy protection for certain applications is a problem when using a public blockchain, whereas keeping accountability and transparency for the transactions is the problem when using a private blockchain. Using cryptographic methods to overcome the privacy protection problem in public blockchains is a solution that increases cost and complexity. Yet, in private or consortium blockchains, [56], centrality can increase and organizations, aggregators or selected set of nodes determine the consensus that becomes permissioned, which contradicts inherent features of a truly decentralized blockchain [33]. In addition, although this trade-off is mostly considered in the privacy area, it is very important that the structure of smart contracts are simple and do not have unnecessary functions in order to avoid cyber security vulnerabilities and financial frauds previously seen in these networks [57]. Therefore, in a hybrid manner both of the two is used to balance this trade-off and outcomes are discussed as well.

On the other hand, by all means proposed framework inherently bears the security precautions and features that are coming to life by virtue of blockchain such as preventing double-spending, keeping transactions in a secure immutable common ledger, authenticating transactions and being in a true P2P manner.

1) RANDOM PARTICIPANTS CAN ATTEND THE AUCTION

It is not a meaningful attempt for the random participant unless it tries to do DoS attack. This problem is solved by selecting the next highest bid as a winner when the winner is not in the physical network. In that case, the fake winner can not withdraw the bid back and that keeps the system secure.

2) PARTICIPANTS CAN SEE OTHERS' BIDS AND DECIDE THEIR STRATEGY

It is assumed that all the participants are honest to reduce the cost of running the smart contract. In this article, the blockchain model is proposed and each module is abstracted to make the platform modifiable easily. When the network is heterogeneous and privacy issue becomes important, the auction module could be supported with cryptographic applications like sealed auctions, or encrypted comparisons to find the highest bid as in [58]–[60]. In that case, the gas price will increase but the system can deal with the privacy issue. Otherwise, as previously mentioned, a hybrid model akin to [56] can be adapted easily by using private blockchain for privacy-driven portions and public blockchain for transparency-driven portions of the proposed platform.

3) GENERAL SECURITY CONCERNS ARE SOLVED BY BLOCKCHAIN PLATFORM

Nobody can bid on behalf of some other nodes, it is not allowed to have double-spending, transactions are kept in a secure distributed database, and smart contract assures the trusted agreements between peers.

4) PARTICIPANTS CAN APPLY DIFFERENT CHARACTERISTICS TO GET THE ADVANTAGE OVER OTHERS AND THAT CAN CAUSE SOME DEADLOCKS

This issue is not about the platform itself, however, the proposed system can apply different penalty schemes on their VPP network. Please note that the proposed architecture is aimed to support VPP network with its peer to peer background. Different characteristics of participants will be analyzed as future work with game-theoretical approaches.

V. CASE STUDIES

The proposed architecture is tested and validated under four different case studies by using a one-day realistic energy data from Australia, Perth Region. Fig. 5 represents the total generation and the total load changes on a specific day. In this figure, the green curve represents the total load, and total production is represented with the orange curve. Also, the yellow curve shows the gap between the total production and total load at a specific time, and the blue curve is the ESS-aided version of the yellow curve in the VPP in 24 hours.

Based on the information given in Fig. 5, it is observed that there could be four different cases that may occur during a day, as shown. In general, there could be two scenarios without ESS: (i) total generation could match with total load, or (ii) VPP needs to feed the system, e.g., trading with market operator, since the load is higher than the production rate. Please note that diesel generator and ESS are also considered as agents in the system. Therefore, they are able to buy/sell energy among all others in the proposed architecture. Thus, it is also needed to add two more cases to determine the roles of the ESS in these two general cases. Table 2 presents a general overview of the case studies whether the one hour backup of ESS is sufficient, the energy demand in VPP is met or not, and the ESS is in charging or discharging status.

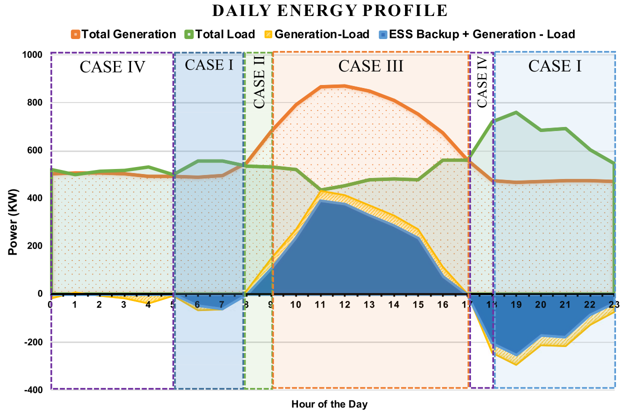


FIGURE 5. Energy profile of the VPP (using realistic data).

TABLE 2. Overview of cases.

Cases	ESS Status	Satisfying Demand	Charging Status
I	short	not enough	discharging
II	short	just enough	charging needed
III	charged	enough	charging
IV	charged	just enough	discharging

A. CASE I

When ESS is short, and the total production is not enough to feed the demand, VPP needs to buy the power from the grid. These hours are represented as *Case I* in Fig. 5. During these hours, there is definitely not enough energy in the system; however, smart-contract must be still active in letting agents get energy from the producers. Since the agents are already out of power, it is better to use RS 3 to avoid the cost of deploying smart contracts. In Table 3 line four, the difference between Total Generation (TG) together with one hour backup of ESS and Total Load (TL) is drastically low that represents the Case I clearly and VPP can not run the system

TABLE 3. Cases during the certain time of the day.

Time	TG (KW)	TL (KW)	TG + Backup - TL	TG - TL	Cases
03:00	502.9069	518.8	0	-15.8930	IV
08:00	551.3342	536.4	0	14.9342	II
13:00	850.8720	479.0	331.8720	371.8720	III
21:00	476.1627	694.6	-178.4372	-218.4372	I

without the help of market operator. Even though total energy is not enough in the VPP, there may be agents exceeded or failed behind their forecasted production. In a penalty condition, peers can trade with each other with the proposed platform. For this specific situation, the system could also allow running RS 2, which gives truly P2P trading among the prosumers in need not to get punished.

B. CASE II

When the total generation is enough to feed the demand, but there is not enough energy in the ESS, the system should also charge the ESS since all of them are in charging mode. *Case II* is represented in Fig. 5, when the yellow part is above zero between 8-9 am. In these hours, the number of trading and transactions occur on the blockchain architecture is increasing. In Table 3 at 8:00, the TG can match the TL but ESS can not sell energy since they need to be charged. In this case, all three schemes are possible, but the first looks centralized, the third one is less secure, and so RS 2 is better to operate.

C. CASE III

The difference between the Case II and Case III is the role of the ESS. In this case, ESS is probably in charging mode, yet they can sell energy as well. During this period, VPP is in islanded mode, and all trading and transactions are handled inside VPP. Thus, VPP could go to offline mode and let all peers manage themselves with the power of the blockchain architecture that eliminates third parties. In Table 3 line three, excessive energy is shown to explain that VPP can work in islanded mode with no doubt. It is recommended to operate RS 2 to be able to trade in a P2P manner and to reduce the communication cost that could happen when RS 3 is used.

D. CASE IV

This is almost the same as Case I, but the role of the ESS is different. The difference between TG and TL is not as low as in Case I and more precisely, at 03:00 the difference between TG with ESS backup and TL is zero in Table 3. ESS has just enough backup to compensate the difference, which means it's in discharging mode. Others need to trade the energy from the ESS, otherwise VPP should feed the system with the help of market operator. Although there is no need to buy energy from the grid, VPP and market operator is still connected, where the blue line touches zero, from midnight to 5 am and at 5 pm. For this scenario, all three RS can work, but it is offered to operate RS 3 since there are a few participants interested in selling energy. P2P_ETC can deploy a contract and modify it when a new one wants to start an auction to avoid the cost of deployment.

E. ANALYSIS

Proper running schemes for the given cases are discussed in this section. A high-level summarized overview and comparisons of running schemes for each case is given in Table 4. The proposed architecture requires P2P_ETC to open repeatedly buying and selling auctions, Running Scheme (RS) 1, for definite time periods regardless of the case to assure connectivity among all the participants. Other schemes can be applied based on the VPP power distribution conditions. There are two important factors while deciding the running schemes: (i) overall demand on the network, and (ii) ESS condition. When there is excess energy, in Case II and Case III, number of auctions will be increased for trading processes. Thus, RS 2 is recommended for both cases, especially for Case III it is strongly recommended to reduce the communication cost and assure truly P2P network. On the other cases, RS 2 is not recommended unless the penalty is not applied for unsuccessful peers that promise to generate a particular amount of energy in Case I. RS 3 is recommended when generated energy is not enough to run the network. Optional schemes can be chosen depending on the requirements and operational conditions.

TABLE 4. Recommended running schemes.

	RS 1	RS 2	RS 3
Case I	Recommended	To Reduce Penalties	Recommended
Case II	Optional	Recommended	Optional
Case III	Optional	Strongly Recommended	Optional
Case IV	Optional	Not Recommended	Recommended

VI. PERFORMANCE EVALUATION

Blockchain-based solutions become hugely influential recently because of its transparent and distributed architecture. Since the technology is quite new, there are different evaluation metrics to measure the proposed system's performance. Each proposed platform can have distinct advantages over other solutions based on these metrics. The overall performance can vary depending on the description of the problem and its aims. According to the used platform, e.g., permissioned blockchain (Hyperledger) with chain code implementation or public Ethereum network with a smart contract, the measured performance metrics can be remodeled.

In this work, smart contract enabled public Ethereum network is introduced, and average gas costs are discussed in Table 1. Some other metrics are recommended to measure the overall performance of the blockchain platforms [61]. Since the proposed system is working on the public Ethereum network, it is not required to test fundamental metrics for the core platform, like transactions per second (tps), which is well-known. Instead, the smart contract's performance on Average Execution Time for different loads and cases are presented in this section. The execution time shows the elapsed time between a transaction request time, $t_{tx_{input}}$, and its confirmation with state updating in the network, $t_{tx_{confir}}$. In order to obtain the elapsed time between a bidding request from an agent to the smart contract and its confirmation notification

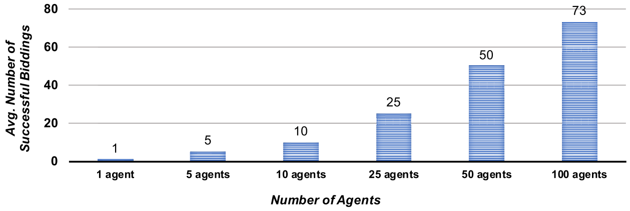


FIGURE 6. The general performance of smart contract under different workloads.

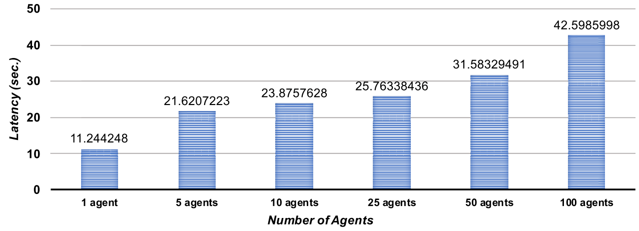


FIGURE 7. Average processing time for bids when different number of agents use the system at the same time.

from the network to the agent, web3.js scripts that we coded were utilized by connecting to the network via 'Infura.io'. The average value is calculated by taking an average for all the requests in a given time span, from t_i to t_j, as shown in (1).

$$AET = \frac{\sum_i^j (t_{tx_{confir}} - t_{tx_{input}})}{Count(tx \text{ in } (t_i, t_j))} \tag{1}$$

The general performance of the platform is measured when 1, 5, 10, 15, 25, 50, and 100 participants are located in the system. Furthermore, to show the case performances, tests are applied at different times. In this framework, the bidding mechanism is proposed for the agreement on the P2P matching process. The system is working on asynchronous mode, and two participants might try to increase the highest bid at the same time, which can cause conflicts. Hence, some of the bidding attempts could be refused when the number of participants increases. In Fig. 6, the average number of successful biddings are presented to show the robustness of the platform. When the system is overloaded with 100 agents, 73 percent of the requests are approved by the smart contract. Fig. 7 shows the average execution time under different workloads. Average execution time is affected by the load, and increasing the number of agents raises the processing time. With this result, 50 seconds is recommended as a period between the following requests from the same participant, to keep the system consistent. The smart contract performance in different cases is also evaluated, without Case II since it is considered a transition period. The results are presented in Fig. 8 for 100 agents which is an overloaded scenario. Since there are many transactions needed to be processed in Case III, the average execution time becomes higher than other cases.

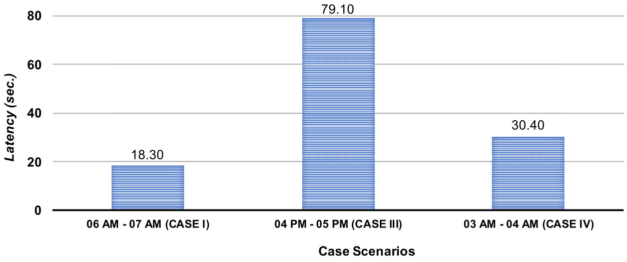


FIGURE 8. Performance of the the system under different case scenarios when it is overloaded with 100 agents.

VII. CONCLUSION

In this work, a blockchain-based bidding platform and cryptographic testing environment have been developed to achieve an efficient, transparent, and economic P2P energy trading within VPP framework using Smart Contract. A public blockchain, unlikely to other applications, is implemented, algorithmic steps are generated, and the usage schemes are discussed in detail. Smart contract development and implementation that facilitates P2P energy trading via auction-based bidding mechanisms are explained including the details of the functions. The proposed auction-based bidding platform interlinks various software, e.g., Solidity, Remix, Metamask, Infuro.io and Ropsten to enable blockchain-based energy trading which works in a real-life cryptographic environment. Possible running schemes are discussed to achieve effective bidding platform to deal with both cost and security concerns. In light of real generation load data from Western Australia, the suitable running scheme(s) for P2P energy trading under the developed platform is demonstrated and suitable recommendations are made.

In order to reach an optimized and efficient operation of the model(s), deep learning and artificial intelligence algorithms may be utilized. Auto-managing tools with deep learning, game-theoretical analysis for profit maximization of VPP, and other security and defense mechanisms are considered as the future tasks before commercializing the developed energy trading platform.

REFERENCES

[1] X. Cao, J. Wang, and B. Zeng, "Distributed generation planning guidance through feasibility and profit analysis," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5473–5475, Sep. 2018.

[2] A. Majzoobi and A. Khodaei, "Application of microgrids in supporting distribution grid flexibility," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3660–3669, Sep. 2017.

[3] A. Khayatian, M. Barati, and G. J. Lim, "Integrated microgrid expansion planning in electricity market with uncertainty," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3634–3643, Jul. 2018.

[4] L. Yavuz, A. Önen, S. M. Muyeen, and I. Kamwa, "Transformation of microgrid to virtual power plant—A comprehensive review," *IET Gener., Transmiss. Distrib.*, vol. 13, no. 11, pp. 1994–2005, Jun. 2019.

[5] C. Zhang, J. Wu, M. Cheng, Y. Zhou, and C. Long, "A bidding system for Peer-to-Peer energy trading in a grid-connected microgrid," *Energy Procedia*, vol. 103, pp. 147–152, Dec. 2016.

[6] W. Saad, Z. Han, H. Poor, and T. Basar, "Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 86–105, Sep. 2012.

[7] I. Atzeni, L. G. Ordonez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Demand-side management via distributed energy generation and storage optimization," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 866–876, Jun. 2013.

[8] Y. Wang, W. Saad, Z. Han, H. V. Poor, and T. Basar, "A game-theoretic approach to energy trading in the smart grid," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1439–1450, May 2014.

[9] N. Zhang, Y. Yan, and W. Su, "A game-theoretic economic operation of residential distribution system with high participation of distributed electricity prosumers," *Appl. Energy*, vol. 154, pp. 471–479, Sep. 2015.

[10] W. Tushar, T. K. Saha, C. Yuen, P. Liddell, R. Bean, and H. V. Poor, "Peer-to-peer energy trading with sustainable user participation: A game theoretic approach," *IEEE Access*, vol. 6, pp. 62932–62943, 2018.

[11] W. Tushar, T. K. Saha, C. Yuen, T. Morstyn, Nahid-Al-Masood, H. V. Poor, and R. Bean, "Grid influenced Peer-to-Peer energy trading," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1407–1418, Mar. 2020.

[12] T. Morstyn, A. Teytelboym, and M. D. Mcculloch, "Bilateral contract networks for Peer-to-Peer energy trading," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2026–2035, Mar. 2019.

[13] T. Morstyn, N. Farrell, S. J. Darby, and M. D. McCulloch, "Using peer-to-peer energy-trading platforms to incentivize prosumers to form federated power plants," *Nature Energy*, vol. 3, no. 2, pp. 94–101, Feb. 2018.

[14] M. F. Zia, E. Elbouchikhi, and M. Benbouzid, "Microgrids energy management systems: A critical review on methods, solutions, and prospects," *Appl. Energy*, vol. 222, pp. 1033–1055, Jul. 2018.

[15] C. Zhang, J. Wu, C. Long, and M. Cheng, "Review of existing Peer-to-Peer energy trading projects," *Energy Procedia*, vol. 105, pp. 2563–2568, May 2017.

[16] *Vandebron*. Accessed: Jun. 9, 2019. [Online]. Available: https://vandebron.nl

[17] *SmartWatt Energy Solutions*. Accessed: Jun. 9, 2019. [Online]. Available: http://www.smartwatt.com

[18] *Smartwatt Solutions for Energy Systems*. Accessed: Jun. 9, 2019. [Online]. Available: https://smartwatt.pt

[19] *Research Project Smart Watts*. Accessed: Jun. 11, 2019. [Online]. Available: https://www.psi-energymarkets.de/en/company/research-and-development/smart-watts/

[20] *Research Project Smart Watts*. Accessed: Jun. 11, 2019. [Online]. Available: https://www.psi-energymarkets.de/en/company/research-and-development/smart-watts/

[21] B. Brandherm, J. Baus, and J. Frey, "Peer energy cloud—Civil marketplace for trading renewable energies," in *Proc. 8th Int. Conf. Intell. Environ.*, Jun. 2012.

[22] *Electron*. Accessed: Jun. 6, 2019. [Online]. Available: https://www.electron.org.uk

[23] LO3 Energy. *The USA's First Consumer Energy Transaction Begins 'Power to the People' Revolution in New York*. Accessed: Jun. 17, 2019. [Online]. Available: https://lo3energy.com/usas-first-consumer-energy-transaction-begins-power-people-revolution-new-york/

[24] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets," *Appl. Energy*, vol. 210, pp. 870–880, Jan. 2018.

[25] *White Gum Valley Energy Sharing*. Accessed: Jun. 17, 2019. [Online]. Available: https://westernpower.com.au/energy-solutions/projects-and-trials/white-gum-valley-energy-sharing/

[26] *Power Ledger*. Accessed: Jun. 17, 2019. [Online]. Available: https://www.powerledger.io

[27] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renew. Sustain. Energy Rev.*, vol. 100, pp. 143–174, Feb. 2019.

[28] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized Peer-to-Peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.

[29] X. Huang, C. Xu, P. Wang, and H. Liu, "Lnsc: A security model for electric vehicle and charging pile management based on blockchain ecosystem," *IEEE Access*, vol. 6, pp. 13565–13574, 2018.

[30] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep. 2018.

[31] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.

[32] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer, and C. Weinhardt, "A blockchain-based smart grid: Towards sustainable local energy markets," *Comput. Sci. Res. Develop.*, vol. 33, nos. 1–2, pp. 207–214, Feb. 2018.

[33] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3548–3558, Jun. 2019.

[34] S. Wang, A. F. Taha, J. Wang, K. Kvaternik, and A. Hahn, "Energy crowdsourcing and Peer-to-Peer energy trading in blockchain-enabled smart grids," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 8, pp. 1612–1623, Aug. 2019.

[35] E. M. Radi, N. Lasla, S. Bakiras, and M. Mahmoud, "Privacy-preserving electric vehicle charging for Peer-to-Peer energy trading ecosystems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[36] H. Liu, Y. Zhang, S. Zheng, and Y. Li, "Electric vehicle power trading mechanism based on blockchain and smart contract in v2g network," *IEEE Access*, vol. 7, pp. 160546–160558, 2019.

[37] M. L. Di Silvestre, P. Gallo, M. G. Ippolito, R. Musca, E. R. Sanseverino, Q. T. T. Tran, and G. Zizzo, "Ancillary services in the energy blockchain for microgrids," *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 7310–7319, Nov. 2019.

[38] M. Li, D. Hu, C. Lal, M. Conti, and Z. Zhang, "Blockchain-enabled secure energy trading with verifiable fairness in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6564–6574, Oct. 2020.

[39] R. Khalid, N. Javaid, A. Almogren, M. U. Javed, S. Javaid, and M. Zuair, "A blockchain-based load balancing in decentralized hybrid p2p energy trading market in smart grid," *IEEE Access*, vol. 8, pp. 47047–47062, 2020.

[40] D. Han, C. Zhang, J. Ping, and Z. Yan, "Smart contract architecture for decentralized energy trading and management based on blockchains," *Energy*, vol. 199, May 2020, Art. no. 117417.

[41] N. Lasla, M. Al-Ammari, M. Abdallah, and M. Younis, "Blockchain based trading platform for electric vehicle charging in smart cities," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 80–92, Jun. 2020.

[42] C. Burger, A. Kuhlmann, P. Richard, and J. Weinmann, "Blockchain in the energy transition. A survey among decision-makers in the German energy industry," *DENA Ger. Energy Agency*, p. 41, 2016.

[43] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: www.Bitcoin.Org

[44] A. S. Musleh, G. Yao, and S. M. Muyeen, "Blockchain applications in smart grid–review and frameworks," *IEEE Access*, vol. 7, p. 86746–86757, 2019.

[45] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[46] M. Mukhopadhyay, *Ethereum Smart Contract Development: Build Blockchain-Based Decentralized Application Using Solidity*. Birmingham, Mumbai: Packt, 2018.

[47] Ethereum Foundation, *Ethereum Project*. Accessed: Jul. 7, 2019. [Online]. Available: https://www.ethereum.org

[48] N. Szabo, "Formalizing and securing relationships on public networks," *1st Monday*, vol. 2, no. 9, Sep. 1997, doi: 10.5210/fm.v2i9.548.

[49] Ethereum Foundation. (2018). *Remix Ethereum Browser IDE*. [Online]. Available: https://remix.readthedocs.io/en/latest/

[50] Ethereum Community. (2019). *TestNet Ropsten (ETH) Blockchain Explorer*. [Online]. Available: https://ropsten.etherscan.io

[51] L. Koçkesen and E. A. Ok, "Auctions," in *An Introduction to Game Theory*. New York, NY, USA: Koç Univ., New York Univ., 2007, pp. 117–130. Accessed: Jun. 17, 2020. [Online]. Available: http://home.ku.edu.tr/lkockesen/public_html/teaching/econ333/lectnotes/uggame.pdf

[52] C.-C. Chang, T.-F. Cheng, and W.-Y. Chen, "A novel electronic english auction system with a secure on-shelf mechanism," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 4, pp. 657–668, Apr. 2013.

[53] Ethereum Foundation. (2019). *Common Patterns, Withdrawal from Contracts*. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.9/common-patterns.html

[54] C. Liu, K. K. Chai, X. Zhang, E. T. Lau, and Y. Chen, "Adaptive blockchain-based electric vehicle participation scheme in smart grid platform," *IEEE Access*, vol. 6, pp. 25657–25665, 2018.

[55] L. Bader, J. C. Burger, R. Matzutt, and K. Wehrle, "Smart contract-based car insurance policies," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–7.

[56] H. Desai, M. Kantarcioglu, and L. Kagal, "A hybrid blockchain architecture for privacy-enabled and accountable auctions," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 34–43.

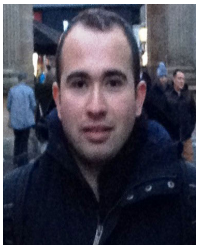
[57] X. Zhao, Z. Chen, X. Chen, Y. Wang, and C. Tang, "The DAO attack paradoxes in propositional logic," in *Proc. 4th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2017, pp. 1743–1746.

[58] N. Y. Lee, J. Yang, M. M. H. Onik, and C. S. Kim, "Modifiable public blockchains using truncated hashing and sidechains," *IEEE Access*, vol. 7, pp. 173571–173582, 2019.

[59] R. Tso, Z. Y. Liu, and J. H. Hsiao, "Distributed E-voting and E-bidding systems based on smart contract," *Electronics*, vol. 8, no. 4, pp. 1–22, 2019.

[60] H. S. Galal and A. M. Youssef, "Verifiable sealed-bid auction on the ethereum blockchain," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2019, pp. 265–278.

[61] P. Zheng, Z. Zheng, X. Luo, X. Chen, and X. Liu, "A detailed and real-time performance monitoring framework for blockchain systems," in *Proc. 40th Int. Conf. Softw. Eng. Softw. Eng. Pract.*, New York, NY, USA, 2018, pp. 134–143.



SERKAN SEVEN received the B.Sc. degree in computer engineering from Kocaeli University, in 2009, and the M.Sc. degree in software engineering from the Queen Mary University of London, in 2013. He is currently pursuing the Ph.D. degree in computer and electrical engineering with Abdullah Gül University. He is working as a Research Assistant at Abdullah Gül University. His current research interests include software engineering, computer networks, and blockchains.

GANG YAO (Member, IEEE) received the M.S. and Ph.D. degrees from the Electrical Engineering Department, Shanghai Maritime University, Shanghai, China, in 2003 and 2008, respectively. In 2008, he joined the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), Nantes, France, as a Postdoctoral Research Fellow. Since the end of 2009, he has been an Assistant Professor with the Electrical Engineering Department, Shanghai Maritime University. His current research interests include multiagent-based intelligent control, fault diagnosis, and faults tolerant control of microgrids, and marine power systems.

AHMET SORAN (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from TOBB ETÜ, Ankara, Turkey, in 2009 and 2012, respectively, and the Ph.D. degree in computer science and engineering from the University of Nevada, Reno (UNR), in 2017. He joined the Computer Engineering Department, Abdullah Gül University, Kayseri, Turkey, in 2018. His research interests include graph theory applications, network communications, device-to-device protocols, network and traffic management, network architecture, smart grids, the Internet of Things, cyber security, and privacy.

AHMET ONEN (Member, IEEE) received the B.Sc. degree in electrical-electronics engineering from Gaziantep University, in 2005, the M.S. degree in electrical-computer engineering from Clemson University, in 2010, and the Ph.D. degree from the Electrical and Computer Engineering Department, Virginia Tech, in 2014. He is currently a Turkish Transmission Network (TEIAS) Academic Advisor. He is working as an Associate Professor at Abdullah Gül University.

S. M. MUYEEN (Senior Member, IEEE) received the B.Sc.Eng. degree from the Rajshahi University of Engineering and Technology (RUET), Bangladesh, formerly known as the Rajshahi Institute of Technology, in 2000, and the M.Eng. and Ph.D. degrees from the Kitami Institute of Technology, Japan, in 2005 and 2008, respectively, all in electrical and electronic engineering. He is currently working as an Associate Professor at the Department of Electrical and Computer Engineering, Curtin University, Perth, WA, Australia. His research interests include power system stability and control, electrical machine, FACTS, energy storage system (ESS), renewable energy, and HVDC systems. He has been a Keynote Speaker and an Invited Speaker at many international conferences, workshops, and universities. He has published more than 225 papers in different journals and international conferences. He has published seven books as an author or an editor. He is a Fellow of Engineers Australia. He serves as an Editor/Associate Editor for many prestigious journals from IEEE, IET, and other publishers, including IEEE Transactions on Sustainable Energy, IEEE Power Engineering Letters, *IET Renewable Power Generation*, *IET Generation, Transmission and Distribution*, and so on.