



## Article

# BPET: A Unified Blockchain-Based Framework for Peer-to-Peer Energy Trading

Caixiang Fan <sup>1</sup>, Hamzeh Khazaei <sup>2</sup> and Petr Musilek <sup>1,\*</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada; caixiang@ualberta.ca

<sup>2</sup> Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada; hkh@yorku.ca

\* Correspondence: pmusilek@ualberta.ca

**Abstract:** Recent years have witnessed a significant dispersion of renewable energy and the emergence of blockchain-enabled transactive energy systems. These systems facilitate direct energy trading among participants, cutting transmission losses, improving energy efficiency, and fostering renewable energy adoption. However, developing such a system is usually challenging and time-consuming due to the diversity of energy markets. The lack of a market-agnostic design hampers the widespread adoption of blockchain-based peer-to-peer energy trading globally. In this paper, we propose and develop a novel unified blockchain-based peer-to-peer energy trading framework, called BPET. This framework incorporates microservices and blockchain as the infrastructures and adopts a highly modular smart contract design so that developers can easily extend it by plugging in localized energy market rules and rapidly developing a customized blockchain-based peer-to-peer energy trading system. Additionally, we have developed the price formation mechanisms, e.g., the system marginal price calculation algorithm and the pool price calculation algorithm, to demonstrate the extensibility of the BPET framework. To validate the proposed solution, we have conducted a comprehensive case study using real trading data from the Alberta Electric System Operator. The experimental results confirm the system's capability of processing energy trading transactions efficiently and effectively within the Alberta electricity wholesale market.

**Keywords:** blockchain; distributed energy resources; smart energy systems; peer-to-peer energy trading; energy market



**Citation:** Fan, C.; Khazaei, H.; Musilek, P. BPET: A Unified Blockchain-Based Framework for Peer-to-Peer Energy Trading. *Future Internet* **2024**, *16*, 162. <https://doi.org/10.3390/fi16050162>

Received: 16 April 2024  
Revised: 30 April 2024  
Accepted: 5 May 2024  
Published: 7 May 2024



**Copyright:** © 2024 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

As distributed energy resources (DERs) such as solar panels, wind turbines, geothermal energy, plug-in electric vehicles, and bioenergy are growing, there is a pressing need for a decentralized and open market to facilitate energy trading between small-scale producers (prosumers) and consumers. Peer-to-peer energy trading (P2P-ET) refers to the direct energy exchange between prosumers and consumers. An example is the exchange of surplus renewable energy originating from DERs, like solar generation in homes, offices, factories, and other establishments [1]. P2P-ET has the potential to enhance the benefits of prosumers and consumers, providing increased flexibility to end-users to access clean energy and facilitating the transition to a low-carbon energy system. P2P-ET also affords advantages to other electricity market participants through lowering peak demand, reducing operational expenses, and enhancing system reliability [2].

A blockchain-based system provides a tamper-proof and decentralized solution to ensure the originality and authenticity of the energy data mathematically. Unlike the centralized system, the blockchain stores data copies in a large number of decentralized ledger nodes geographically distributed worldwide. Therefore, blockchain technologies have attracted a lot of attention from researchers and industry practitioners to resolve trust and information security issues in smart grids [3,4].

However, developing such a system entails substantial effort and cost. Challenges include but are not limited to technical complexity, blockchain platforms' scalability and interoperability, regulatory frameworks and standardization, smart contract vulnerabilities, and market dynamics [5,6]. On one hand, the intricacies of the blockchain technology stack demand experienced developers to ensure system security and efficiency. The integration of blockchain into the energy sector requires developers to have a comprehensive understanding of blockchain consensus, cryptographic protocols, smart contract security best practices, and energy trading business logic [7]. On the other hand, the diversity of the energy markets poses challenges in directly reusing existing solutions. Each project must commence from scratch, including deploying a blockchain network, designing smart contracts, and developing the front-end web application. This challenge hinders the widespread adoption of blockchain-based P2P-ET systems globally.

To address this challenge, we have devised a unified and extensible blockchain-based P2P-ET framework. The performant and versatile Enterprise Ethereum client Hyperledger Besu is selected to construct the blockchain infrastructure. Its modular design facilitates the integration of various consensus algorithms, including Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA), i.e., Clique, IBFT2.0, and QBFT. Microservices have been implemented to ensure the system's reliability. The objective of this framework is to empower developers to swiftly create localized P2P-ET systems and explore the potential of blockchain technology in fostering sustainable and efficient energy management practices.

The main contributions of this work are summarized as follows:

- We propose a novel unified blockchain-based P2P-ET framework that facilitates diverse energy trading markets at distribution and end-user levels. It enables developers to plug in customized market rules and swiftly create localized P2P-ET systems.
- Using the proposed framework, we create modular smart contracts based on a local energy market mechanism and build a decentralized application for P2P-ET in the Alberta wholesale market.
- We conduct a case study using real data from the AESO to demonstrate the feasibility and efficiency of the proposed BPET system.

The remainder of this paper is organized as follows. In Section 2, we review the related work on blockchain-based P2P-ET. In Section 3, we analyze current energy markets and energy trading participants, followed by introducing the overview of the proposed BPET framework. Section 4 outlines the BPET system design including energy trading models, smart contracts, and system functionality design. In Section 5, we present technical details of the system implementation. In Section 6, a case study of the wholesale electricity market in Alberta, Canada, is conducted on the developed BPET system using real data from the AESO. In Section 7, we discuss the scalability, security, and potential improvement solutions of the proposed BPET system. Section 8 concludes the article and outlines potential future work.

## 2. Related Work

In recent years, blockchain-based P2P-ET solutions have been extensively studied from various focused perspectives. These include security and privacy [8,9], performance [10,11], system modeling, design and development [12–22], and resolving challenges in specific application scenarios [23–27].

Aitzhan et al. [8] devised and prototyped a decentralized blockchain-based system using digital multi-signature scheme and anonymous encrypted messaging streams to enhance security and privacy in energy trading transactions. In another work, Gai et al. [9] presented a consortium blockchain-oriented approach to resolve the privacy leakage problem.

Abdella et al. [10] proposed a unified blockchain-based energy trading architecture, called UBETA, considering three typical markets: bilateral market, pool market, and balancing market. Similarly, Lohachab et al. [11] proposed a Hyperledger Fabric-based framework for Peer-to-Peer Energy Trading (HFPET) among PHEVs and conducted performance evaluation under different system parameters.

Hahn et al. [12] designed a decentralized architecture for a transactive energy auction on campus based on the Vickrey auction mechanism. Faizan et al. [13] devised a blockchain-based microgrid energy market model and developed smart contracts to enable P2P-ET in both bilateral and pool markets. In another work [14], the consortium blockchain was used to design a hybrid P2P energy trading market where electricity consumers and prosumers trade electricity with one another and the main grid. AlSkaif et al. [15] proposed two strategies, i.e., supply-demand matching and distance-based matching, to implement bilateral trading coefficients in a blockchain-based P2P energy market. Yang et al. [16] used a PoS public blockchain to design a pricing scheme for the P2P-ET market. Mehdinejad et al. [20] created and modeled a completely decentralized P2P energy token market for small-scale producers and consumers within a smart grid context, employing blockchain technology alongside demand response (DR) programs and demurrage mechanisms. In recent works [21,22], authors used the double auction mechanism to calculate energy price.

Li et al. [23] devised a blockchain-based framework for peer-to-peer energy trading for three typical energy trading scenarios in industrial IoT: microgrids, energy harvesting networks, and vehicle-to-grid networks. Kang et al. [24] proposed a localized peer-to-peer (P2P) electricity trading system with consortium blockchain, called PETCON, for plug-in hybrid electric vehicles (PHEVs) in smart grids. Another similar work [27] presented a consortium blockchain-based truthful data trading (D-trading) and energy trading (E-trading) model in the Internet of Electric Vehicles (IoEV). Chaudhary et al. [25] combined blockchain and software-defined networking (SDN) to design a secure energy trading scheme called BEST for EVs. Sheikh et al. [26] focused on the transaction security and Byzantine fault tolerance of energy trading between EVs and the distribution network (DN).

Most previous studies have defined specific market rules within particular application scenarios, along with associated price formation mechanisms. However, many of these solutions have remained theoretical, and no functional systems were developed to execute these market rules. A brief comparison between our work and the previous studies is presented in Table 1.

**Table 1.** A comparison between previous studies and our work.

Paper	Scenarios	Price Mechanism	Payment Design	Implement	Performance Evaluated
[8]	smart grids	negotiation-based mechanism	X	X	X
[12]	campus market	Vickrey auction	✓	✓	X
[23]	microgrids, harvesting, V2G	system-determined	✓	X	X
[24]	plug-in hybrid electric vehicles	iterative double auction	✓	X	X
[14]	P2P and P2G	double closed book auction	✓	✓	X
[25]	electric vehicles	dynamic pricing	✓	X	X
[13]	microgrid	negotiation and double auction	✓	✓	X
[10]	wholesale (bilateral, pool, balancing)	negotiation and double auction	✓	✓	✓
[11]	plug-in hybrid electric vehicles	demand requirement function	X	X	✓
[16]	microgrid	inverse proportion equations	✓	✓	X
[20]	smart grid	bilateral negotiation	X	X	X
[21]	local microgrid	double auction	X	✓	X
[22]	microgrids	double auction	X	X	X
Our work	customizable	customizable	✓	✓	✓

Compared to prior research, our proposed BPET system introduces an end-to-end unified blockchain-enabled P2P-ET solution, allowing developers to customize market rules and price formation mechanisms. The implemented system is evaluated using real-world data, offering valuable performance insights for industry practitioners.

### 3. A Unified Framework

In practice, energy trading occurs across various levels involving corresponding stakeholders within different energy markets. Within an energy distribution network, generators sell energy to electricity retailers and large consumers in a well-structured wholesale market in a peer-to-peer way. At the end-user level, prosumers and consumers directly trade energy in a local market within a microgrid. For instance, residents of smart communities may sell excess electricity generated by solar panels to their neighbors. A microgrid connects to the main grid and participates in the wholesale market as a single entity. The microgrid first self-suffices its local residents then sells its surplus aggregated energy in the wholesale market or purchases energy from it.

#### 3.1. Markets Analysis

From the perspective of energy trading business models, there are three typical market models: bilateral market, pool market, and balancing market [10]. A bilateral market requires a pair of buyers and sellers to negotiate with each other to agree on the final energy price and amount periodically. Buyers and sellers usually negotiate price and quantity off-chain and then make transactions on-chain. A pool market requires buyers and sellers to put bids in a pool, which determines the price and trading volume based on the demand and supply curves. The merit order effect trading model is usually used as a pricing strategy in a pool market, which can be easily implemented in a smart contract [10] and executed on-chain. A balancing market exists to supplement bilateral and pool markets and ensure total power demand and supply are balanced at all times in physical energy networks.

From the perspective of market structures, all P2P-ET markets can be divided into three types: full decentralized market, community-based market, and composite market [28]. A full decentralized market provides a platform for participating prosumers to independently and directly negotiate with each other to decide on the energy trading parameters without centralized supervision. A community-based market [29–31] relies on a community manager to enable members to trade their energy within the community. A composite market is a combination of fully decentralized and community-based markets in which each community and each consumer can interact with each other while maintaining their market properties.

#### 3.2. Participants Analysis

There are essentially four main types of market participants in a blockchain-based energy trading system. The producers or prosumers provide energy to the market. They register basic supplier information (e.g., asset identification, supply capacity, and energy type), submit offers to the market (e.g., unit price and available quantity), and sell energy over a time interval. The consumers bid energy from the market. They register basic consumer information (e.g., asset identification and maximum load), place a bid including the type and amount of energy they want, and consume the energy they bought from the market. The market administrator designs and implements the market rules in smart contracts. This role is usually taken by a selected and experienced individual or by skilled individuals such as system controllers and energy coordinators. Different types of participating entities with authority or reputation, such as distribution system operators (DSOs), transmission system operators (TSOs), market operators (MOs), regulators, and utility companies, can run blockchain nodes in a local market to maintain a permissioned network.

#### 3.3. BPET

An energy trading system must eliminate any single point of failure and cryptographically store all transaction data in a decentralized manner to ensure tamper-proofing and achieve high performance and scalability. Additionally, extending to new markets should be easily attainable with minimal modifications. To fulfill these requirements, we propose a unified and extensible blockchain-based P2P-ET framework by integrating microservices with blockchain technologies, as illustrated in Figure 1. Microservices represent an architec-

tural and organizational approach to software development, where software is composed of small independent services that communicate over well-defined Application Programming Interfaces (APIs).

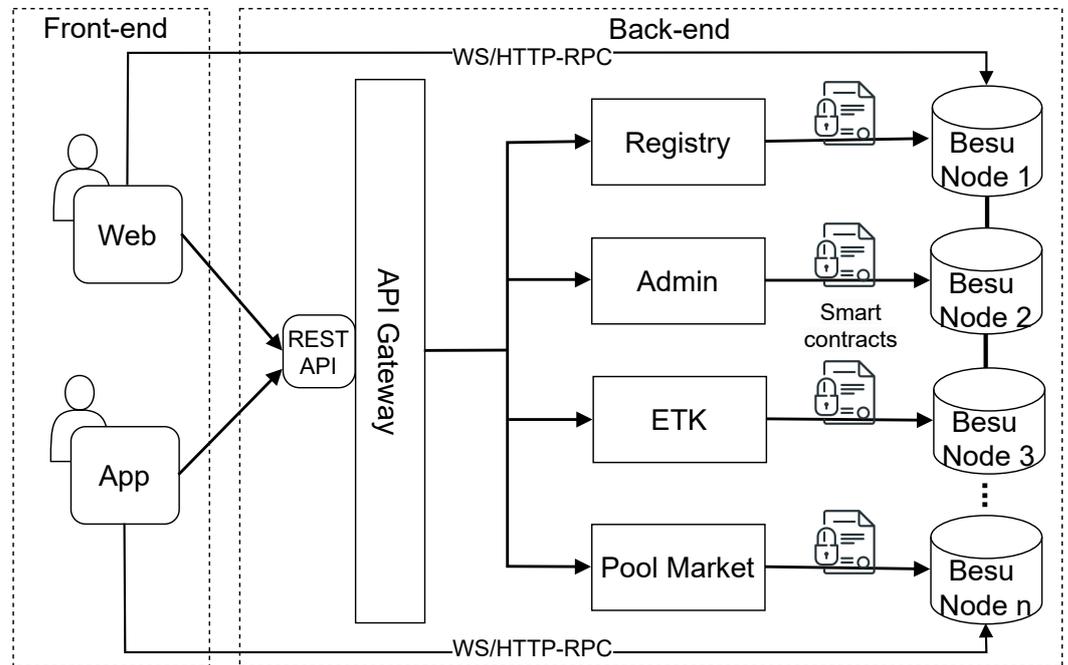


Figure 1. BPET architecture.

We employ the separation of concerns (SoC) design principle to construct the system by separating the front-end and back-end systems. In the proposed solution, all data are stored on-chain, and users interact with the back-end system through a front-end web application with a wallet plugin (e.g., MetaMask) installed. The wallet is responsible for managing accounts and signing transactions. The front-end web communicates with the blockchain in two different ways. When users want to write data to the blockchain and change blockchain states, the web application directly sends the corresponding transactions to the blockchain through the WebSocket or HTTP RPC protocol. When users want to read data from the blockchain or query the state, the web service/application calls the REST API through the API gateway, which aggregates all microservices in the back end.

#### 4. System Design

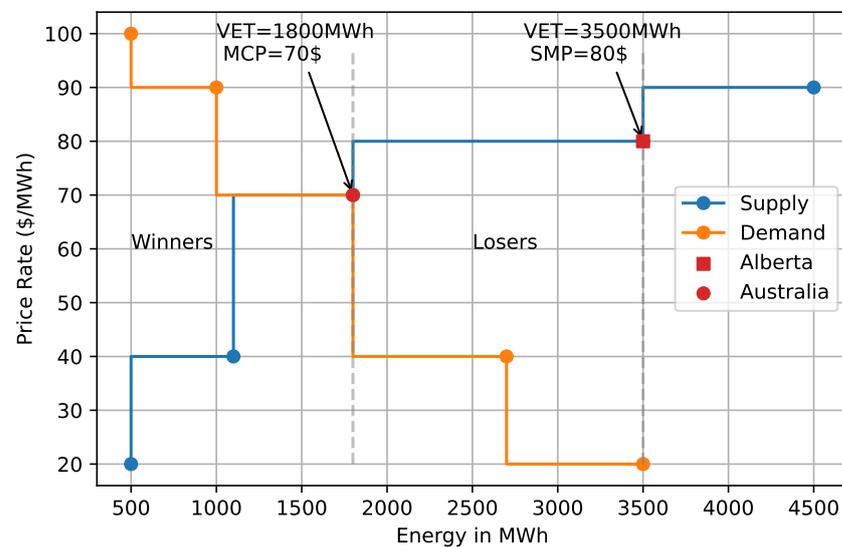
In this section, we introduce more technical details of the proposed BPET system, including energy trading models, smart contract design, and system workflows.

##### 4.1. Energy Trading Models

In the energy industry, the merit order effect [32] is the most widely adopted trading model in pool markets. It allocates electricity generation sources according to their variable costs, favoring the cheapest and most efficient generators. However, the detailed wholesale market rules may vary significantly in practice. For example, while both the wholesale electricity markets in Australia [33] and Alberta [34] employ the merit order effect to determine pool prices, they do so in distinct ways. Figure 2 shows their different market clearance points on the demand–supply curves based on the same sample bidding data in Table 2.

**Table 2.** Sample bidding data of merit order effect.

ID	User Type	Price (\$/MWh)	Aggregated Amount (MWh)
1	Supplier	20	500
2	Supplier	40	1100
3	Supplier	70	1800
4	Supplier	80	3500
5	Supplier	90	4500
10	Consumer	100	500
9	Consumer	90	1000
8	Consumer	70	1800
7	Consumer	40	2700
6	Consumer	20	3500



**Figure 2.** Sample demand–supply curves with different market clearance points for Australia and Alberta Canada.

In the Australian electricity pool market, both consumers and producers compete to win the auction. The market clearance price (MCP) is calculated as the intersection point of the supply and demand curves, which present the aggregated supply or demand on the horizontal axis and the price rate at which participants wish to sell or buy on the vertical axis. Individual bids and offers are matched against the MCP to identify the winners, who are those consumers offering a price rate exceeding the MCP and producers with a bidding price below the MCP. The Australian Energy Market Operator (AEMO) employs other markets (i.e., bilateral and balancing markets) to meet the needs of those bidding losers in the pool market.

In the Alberta electricity pool market, for each hour, market participants submit supply offers and demand bids for electric energy at day-ahead prices for the following day. These offers and bids are organized in a price ascending list called the merit order. The system controllers in the Alberta Electric System Operator (AESO) use the merit order to balance the electricity supply and find a system marginal price (SMP). In this way, the AESO ensures that Alberta internal load (AIL) is met by the most competitively priced electricity.

Typically, the merit order mechanism consists of two steps. The first step is to calculate the aggregated demand–supply curves. Supply offers are sorted by price in ascending order, and demand bids are sorted in descending order. Then, the cumulative summation operation is applied to each list separately to obtain the aggregated supply and demand

amounts in megawatt-hour (MWh). The second step is to determine the pool price and trading volume. These steps differ from each other according to different market price mechanisms. In Australia, the intersection point of the demand and supply curves is determined by matching the price and aggregated energy volume. The pool price MCP and volume of energy to be traded (VET) in a trading interval are defined as:

$$MCP = \{a | a = b \wedge ESA_a = EDA_b\} \tag{1}$$

$$VET = \{ESA_a | a = b \wedge ESA_a = EDA_b\} \tag{2}$$

where  $ESA_a$  is the energy supply aggregate at price  $a$  and  $EDA_b$  is the energy demand aggregate at price  $b$ .

In Alberta, the intersection point is determined by matching the AIL demand and the aggregated supply, which does not consider the bidding price of the buyers. The system marginal price (SMP) and VET for a single minute and the pool price (MCP) for each trading interval, i.e., one hour, are defined as:

$$SMP = \{a | ESA_a \geq AIL\} \tag{3}$$

$$VET = AIL = \sum_{i=1}^m LAB_i \tag{4}$$

$$MCP = \frac{\sum_{i=j}^n SMP_j * d_j}{60} \tag{5}$$

where AIL is the total internal load of Alberta, LAB represents the load amount of bid  $i$ , and there are  $m$  bids at the current minute,  $d_j$  represents the duration (in minutes) of  $SMP_j$  ( $1 \leq j \leq n$ ), and there are  $n$  different marginal prices at the current hour. Table 2 shows the aggregate amounts of supply and demand at each price rate. Two intersection points, Australia (70, 1800) and Alberta (80, 3500), are highlighted in bold.

#### 4.2. Smart Contracts Design

Smart contracts play a critical role in implementing the back-end business logic, such as user registration, transaction management, market rules, and payment settlement. To achieve better extensibility and make the system support different markets, we adopt a modular smart contract design. The core smart contracts and their interactions are shown in Figure 3.

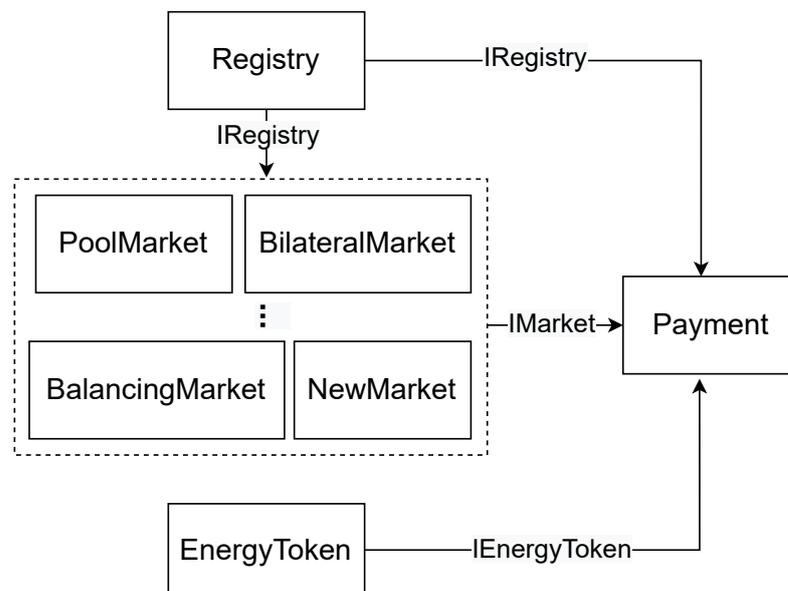


Figure 3. BPET smart contracts design.

Different contracts interact with each other to exchange information through implementing interfaces. Registry provides all functions regarding user management, such as supplier/consumer registration, information updating, and deletion. PoolMarket defines the detailed market rules and records energy transactions. It determines the basic information required to submit an offer/bid and, more importantly, how the price is calculated according to associated market rules and auction models. This contract relies on the Registry smart contract to check user registration status. More market smart contracts (e.g., Bilateralmarket, BalanceMarket, and NewMarket) can be easily integrated into the system by properly implementing the interfaces, i.e., IMarket. This design limits the market type to order book markets, which necessitate registration and input of all demand and supply to calculate trading prices. Further modifications are required to support other market types, such as automated market makers and Oracle-based price mechanisms. EnergyToken provides a unified ERC20 [35] token as the payment method to exchange energy. We design the token as one type of stablecoin similar to the widely used Tether (USDT), USD Coin (USDC), and Binance USD (BUSD) in the current cryptocurrency market. Payment executes the payment process after each trading. It relies on the finalized pool price rate, the total dispatched energy quantity in the PoolMarket, and the payer’s registration status in the Registry.

#### 4.3. System Workflows

Figure 4 shows the system sequence diagram (SSD) of offer submission. First, the user inputs offer information in the front-end web form, which checks if the account has already been registered to prevent the BPET system from denial-of-service (DoS) attacks. Then, it validates if the input amount of energy exceeds the capacity of this provider, the price rate is legitimate, the energy block number is not out of range, etc. Only after all validations pass will the front end send a transaction request to the blockchain. Finally, all blockchain validators reach a consensus to confirm the transaction and send a successful confirmation message to the end user. The front end displays a corresponding error message to the user if any item fails validation. The process of submitting a bid shares substantial similarities with that of submitting an offer. The only difference exists in the validation items. Bid submission validates account registration, submitted energy load, and price rate.

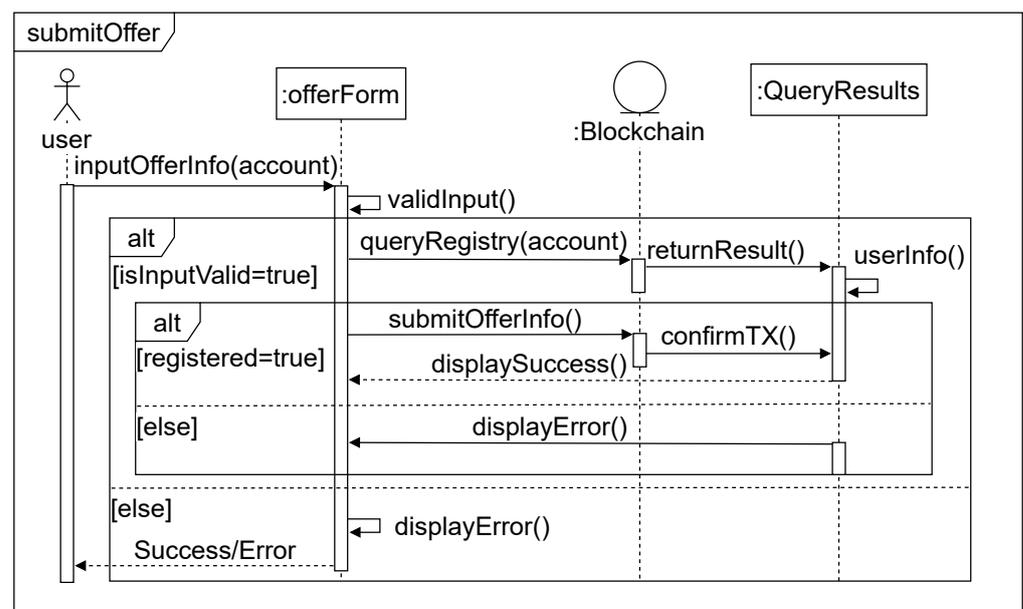


Figure 4. BPET submitOffer sequence diagram.

Other processes include registration, buying or redeeming energy tokens, administrator displaying registered participants, and a portal showing current and historical offers

and bids. They all follow a similar sequence as depicted in Figure 4, in which the front end first validates the input and responds to the user by sending a transaction or displaying an error message according to the validation result. All account access control rules and value validity rules are defined in the blockchain state declarations in smart contracts [36].

## 5. System Implementation

This section introduces more technical details, such as algorithms and techniques used in system implementations.

### 5.1. Smart Contracts

We leverage the price mechanism in the Alberta electricity wholesale market to design smart contracts. In the pool market smart contract, the core business logic determines the market clearance price, called the pool price in Alberta. According to the official guidance documentation [34] on the AESO website, the pool price is determined in the following way. First, the market collects all near-future electricity supplies and demands in terms of price, amount, etc. Pool market participants regularly submit offers and bids to the market for each hour on a day-ahead basis. These supply offers and demand bids are sorted into merit order for each hour of the day. In this way, the AESO ensures that the most competitively priced electricity meets Alberta's overall electricity needs.

Then, the pool price rate, i.e., the dollar cost of a megawatt hour of electricity at the end of a given hour that retailers pay to electricity generators, is calculated from the merit order. Setting the pool price is highly detailed and can be summarized into two steps. The first step is to calculate the system marginal price (SMP), as described in Algorithm 1. The algorithm obtains the latest total demand and energy supply offers as inputs every minute, and it obtains a merit order of all valid offers at this minute. Then, the highest-priced supply offer is calculated by comparing the aggregated supply with the total demand in a loop, where the algorithm automatically dispatches all lower-priced offers. The highest-priced offer submitted to the market, and dispatched by the algorithm, is designated as the system marginal offer (SMO) for this minute, with its price designated as the SMP. All SMOs are stored on-chain and are indexed by their timestamps in minutes.

---

#### Algorithm 1 System Marginal Price Calculation Algorithm

---

**Input:** *bidUpdated, energyOffers* ▷ read from states  
**Output:** *smp* at a particular minute timestamp ▷ update to states

- 1: **if** *bidsNotEmpty* & *offersNotEmpty* **then**
- 2:     *totalDispatched* ← 0 ▷ initiate total dispatched
- 3:     *latestTotalDemand* ← *getLatestTotalDemand()* ▷ get total demand
- 4:     *meritOrderOffers* ← *getMeritOrderSnapshot()* ▷ get a merit order snapshot
- 5:     *currHour* ← *block.timestamp* in hour
- 6:     *N* ← *meritOrderOffers.length*
- 7:     **for** *k* ← 0 to *N* **do**
- 8:         *id* ← *meritOrderOffers<sub>k</sub>*
- 9:         *offer* ← *energyOffers[id]* ▷ get an offer
- 10:         *amount* ← *offer.amount*
- 11:         *account* ← *offer.supplierAccount*
- 12:         *totalDispatched* ← *totalDispatched* + *amount*
- 13:         *dispatchedOffer* ← *DispatchedOffer(account, amount, time)*
- 14:         *dispatchedOffers[currHour].add(dispatchedOffer)*
- 15:         **if** *totalDispatched* ≥ *latestTotalDemand* **then** ▷ meet total demand
- 16:             *currMinute* ← *block.timestamp* in minute
- 17:             *smpOfferIDs[currMinute]* ← *id* ▷ store system marginal offer
- 18:             *smMinutes.add(currMinute)* ▷ store current minute as index
- 19:             **break**
- 20:         **end if**
- 21:     **end for**
- 22: **end if**

---

The second step is to leverage the SMPs from the first step to calculate the pool price, as described in Algorithm 2. At the beginning of each hour, this algorithm calculates the SMP for the following market clearance hour, i.e., the first SMP to calculate the next pool price at the beginning of the next hour. Then, it calculates the pool price as the average of these one-minute SMPs in a loop. For instance, if there is an SMP calculated for each minute in an hour, the pool price will be the average of all the 60 SMPs. However, in most cases, the pool price is calculated as a weighted average (i.e., as a sum of the multiplication results of each SMP and its duration in minutes divided by 60) as shown in Equation (5). All pool prices are stored on-chain and indexed by their timestamps in hours. The SMP is posted to the front-end web application in real time, and the pool price is posted after the hour's end to calculate the total amount of spending/earning and facilitate the payment settlement process.

---

#### Algorithm 2 Pool Price Calculation Algorithm

---

**Input:** *hour, energyOffers*

**Output:** pool price at a particular hour timestamp

```

1:  $H \leftarrow 3600$  ▷ 1 h = 3600 s
2:  $M \leftarrow 60$  ▷ 1 h = 60 min
3:  $sumPrice \leftarrow 0$  ▷ initiate sum price
4:  $N \leftarrow smMinutes.length$  ▷ get the number of system marginal offers
5: for  $k \leftarrow 0$  to  $N$  do
6:    $T \leftarrow smMinutes_k$ 
7:   if  $T \geq hour$  and  $T < hour + H$  then ▷ check offer within previous hour
8:      $id \leftarrow smOfferIDs_T$ 
9:      $offer \leftarrow energyOffers_{id}$ 
10:     $price \leftarrow offer.price$ 
11:     $duration \leftarrow 0$ 
12:    if  $k < N - 1$  and  $smMinutes_{k+1} < hour + H$  then
13:       $duration \leftarrow (smMinutes_{k+1} - smMinutes_k) / M$ 
14:    else
15:       $duration \leftarrow M - (smMinutes_k - hour) / M$  ▷ last offer duration
16:    end if
17:     $sumPrice \leftarrow sumPrice + price \times duration$ 
18:  end if
19: end for
20:  $poolPrice \leftarrow sumPrice / M$  ▷ weighted average system marginal price
21:  $poolPrices[hour] \leftarrow poolPrice$  ▷ store pool price
22:  $poolPriceHours.add(hour)$  ▷ store current hour as index
23: return  $poolPrice$ 

```

---

#### 5.2. Microservices

We selected NestJS, a Node.js framework with built-in support for microservice architecture, to develop the back-end microservices, including Registry, Admin, Energy Token (ETK), and PoolMarket. A microservice is fundamentally a NestJS application that uses a transport layer different from HTTP. Nest supports several built-in transport layer implementations, called transporters, e.g., TCP, Redis, MQTT, RabbitMQ, and Kafka, responsible for transmitting messages between different microservice instances. For simplicity, we choose the default transporter TCP as the communication protocol. Most transporters support two types of message patterns: request–response and event-based message styles. The request–response method is ideal for exchanging messages between services. In contrast, the event-based message pattern is more suitable when a service just wants to publish events without waiting for a response.

To balance the complexity of management and the granularity of the service, we implemented several related functionalities in one microservice.

- Admin Service aggregates operations that require the contract deployer's permission. These include `getSuppliers()`, `getConsumers()`, `approve(approveRequest)`, `requestToken(tokenRequest)`, `calculateSMP()`, and `calculatePoolPrice()`.
- Registry Service provides services related to user registration, including `isRegisteredSupplier(account)`, `isRegisteredConsumer(account)`, `getSupplier(account)`, `getConsumer(account)`, `registerSupplier(supplierRegistryRequest)`, and `registerConsumer(consumerRegistryRequest)`.
- PoolMarket Service provides all query services related to a market, including `getsmp()`, `getProjectedPoolPrice()`, `getMyBids(bidData)`, `getDispatchedOffers()`, `getSystemMarginalMinutes()`, `getMarginalOffer(timestamp)`, `getTotalDemand(timestamp)`, and `getMinMaxPrices()`.
- ETK Service provides services concerning energy token operations, including `getETCOwnerAddress()`, `getBalance(account)`, and `allowance(allowanceRequest)`.
- Gateway Service acts as a client proxy, which aggregates all microservices functionality and provides a unified REST API portal to the front end and developers.

These microservices enhance the reliability of the BPET system in two ways. First, the loose coupling structure assigns each service a specific task, allowing them to work relatively independently. If one service fails, its impact on others is limited, thus preventing catastrophic system failure. Second, container orchestration tools like Docker Swarm and Kubernetes provide high availability through a failover mechanism. They deploy multiple services as a cluster for an application and automatically switch to the backup service when the leader fails.

### 5.3. Web Application

We implement the front end using the popular open-source web development framework Next.js, which enables React-based web applications with server-side rendering and generating static websites. JavaScript and Google's React component library Material UI (MUI) are used to develop all web pages. MetaMask is installed as a plugin to the browser and connected to the blockchain network through the HTTP-RPC protocol. Ethers.js library is used to interact with smart contracts. All developed back-end microservices and the front-end web service are containerized and deployed on the server using Docker Compose to achieve better scalability.

## 6. Case Study

In this section, we take the case of the wholesale electricity market in Alberta to study the effectiveness and efficiency of the proposed BPET system. As stated on its website, a fundamental principle of the Alberta electricity system is that supply and demand must be perfectly matched at all times. In the proposed BPET system, the fundamental principle remains, while a complete electricity trading workflow is simplified into four main stages: initiation, buying ETK, trading energy, and payment.

### 6.1. Historical Data Analysis

AESO provides many public data sets such as merit order lists, pool prices, and marginal prices. To keep the seasonality property of the real data, we select a whole year's historical data of merit orders and marginal prices from September 2021 to August 2022 to statistically analyze the total number of participants and the expected offer and bid submission TPS.

From the historical merit order data set, we identify 201 unique assets, all of which are energy suppliers or producers. Since AESO does not provide information on individual electricity consumers, we obtained 60 retailers from the *Retailers and Distributors* website [37] as consumers in the Alberta electricity wholesale market. Therefore, there are 261 participants registered at the initiation stage. Consequently, there are at most 261 token transfer transactions before trading. Both registration and buying of energy

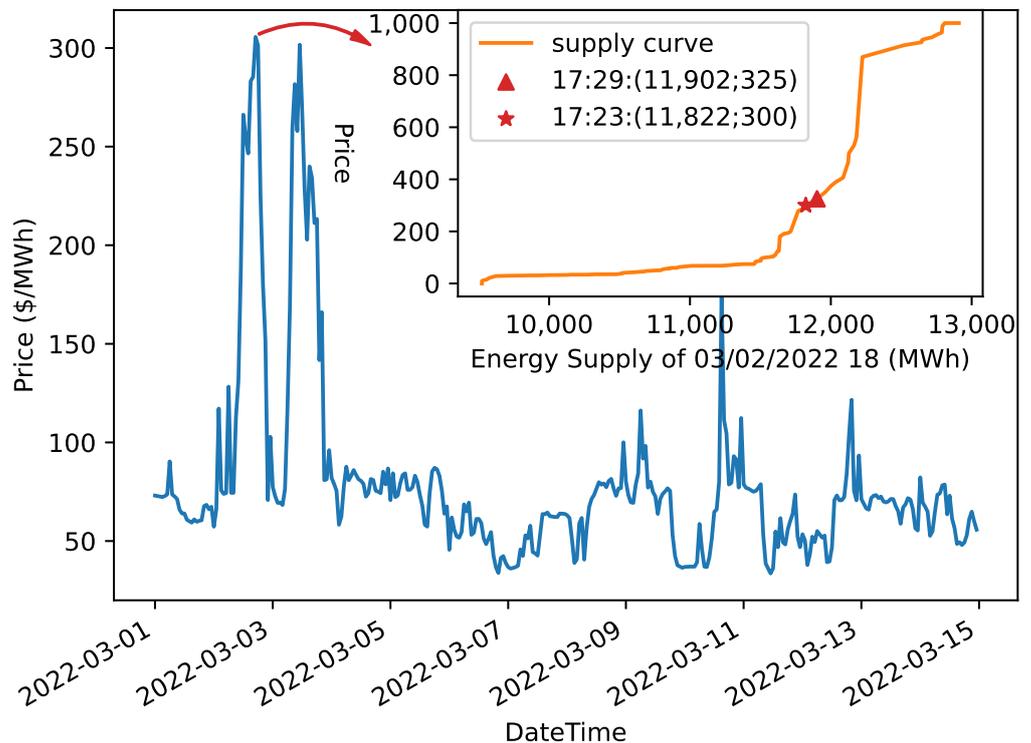
tokens require BPET to process transactions asynchronously, which does not necessitate concurrent high performance.

In the trading stage, offers usually do not have frequent updates after initial submissions. An offer is updated only if it does not appear in the previous hour’s offer list but arrives in the current hour. The comparison factors are listed in a tuple (Asset Id, BlockNumber, Price, Size, Available MW). For simplicity, we only submit updated offers in the next hour. Therefore, the offer submission rate equals the offer update rate, which also applies to the bid submission. Throughout the entire year, the most frequent offer updates occur at the first hour of the day, with a maximum of 245, a minimum of 52, and a mean of 106.

Each time the total demand changes, the market calculates a system marginal price (SMP). Thus, we estimate the demand update rate as the number of SMP calculations each hour. It is worth noting that the total demand changes are measured in MWh. By analyzing the historical SMP data set, we find that all hours have a close demand update rate, with a maximum of 17, a minimum of 1, and a mean of around 4. This indicates that the market system calculates a maximum of 17 SMPs in one hour.

### 6.2. System Simulation

Figure 5 presents the simulation pool prices using the first weeks of AESO energy data in March 2022. The simulation results indicate that the BPET system can calculate the pool prices according to the given total AIL and supplies. The spike points on March 2nd at the hour ending of 18 happened due to the demand increasing at a limited supply level, which made the system’s marginal prices relatively high throughout the whole hour. This situation also happened on 3 March 2022.



**Figure 5.** BPET simulations with the AESO energy data in the first two weeks of March 2022.

### 6.3. System Performance Evaluation

We leveraged Hyperledger Caliper v0.4.2 with Ethereum SDK v1.4 as the benchmark tool to evaluate the blockchain system. In each test, Caliper was running in a container on a client VM with rich resources (e.g., 16 vCPU and 60 GB RAM). The results were

stored in performance reports and then processed using Pandas 2.2.2, Numpy 1.26.4 and Matplotlib 3.8.4.

### 6.3.1. Experimental Setup

We performed the experiments on an OpenStack cloud environment [38]. All the network and blockchain parameters were the same as the baseline configuration of our previous work [39].

### 6.3.2. Workloads

Table 3 shows an overview of the workloads used in performance benchmarking. To comprehensively benchmark the smart contracts, we selected their core public functions to generate workloads. Workload modules were customized by creating a specific JavaScript file and defining transaction workloads with necessary parameters to make Caliper work with the energy trading smart contracts. The transaction rate for each operation varied from 10 TPS to 80 TPS. For each type of transaction, we set up the total number of transactions to be the same as the number of workers, which will be sent at a fixed rate in a single test and result in one transaction per worker. We ran five replicas for a single test under a specific transaction rate, generating five performance reports. After deploying all smart contracts to the Hyperledger Besu network, we ran tests in the order of Registry, EnergyToken, and PoolMarket.

**Table 3.** Overview of benchmarking workloads.

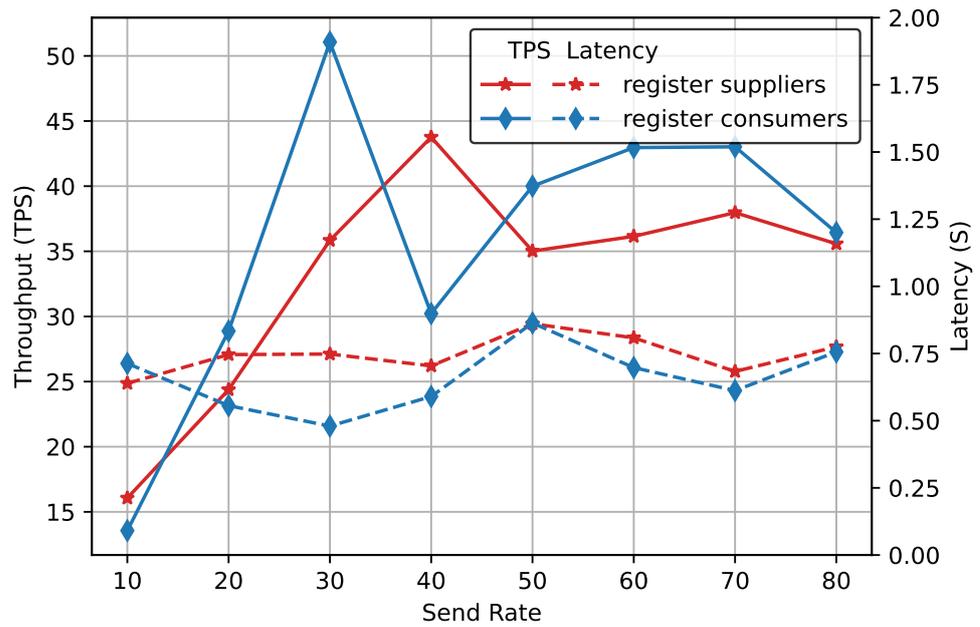
Contracts	Operations	Send Rates	txNum
EnergyToken.sol	transfer	10 → 80 TPS	1 tx/worker
	query		
PoolMarket.sol	submitOffer		
	submitBid		
Registry.sol	registerSupplier		
	registerConsumer		

### 6.3.3. Performance Metrics

In the experimental study, we evaluate two basic performance metrics: throughput and latency. Throughput refers to the number of transactions that a blockchain network processes within a designated timeframe, usually measured in transactions per second (TPS). Latency is the total amount of time between submitting a valid transaction or a query request to the network and the time the network has confirmed the transaction or returned the response, usually measured in seconds.

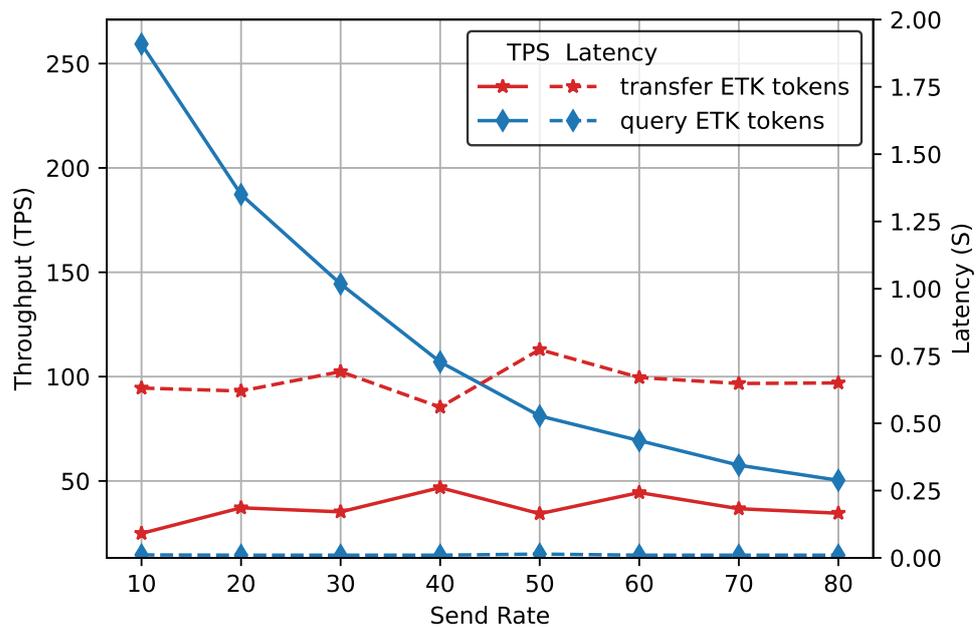
### 6.4. Evaluation Results

Figure 6 shows the benchmark results of the Registry contract. We observe that the throughput of both registration operations increases near-linearly and then drops to a steady level of around 40 TPS as the transaction send rate increases from 10 TPS to 80 TPS. The throughput of registering consumers is slightly higher (around 10% in most cases) than registering suppliers, which is expected because registerSupplier stores one more parameter blockAmount than registerConsumer, and it includes an additional step of updating the total supply capacity for each registered supplier. Accordingly, the latency of registerConsumer is slightly lower than that of registerSupplier. But, both remain at a low level of around 0.75 s.



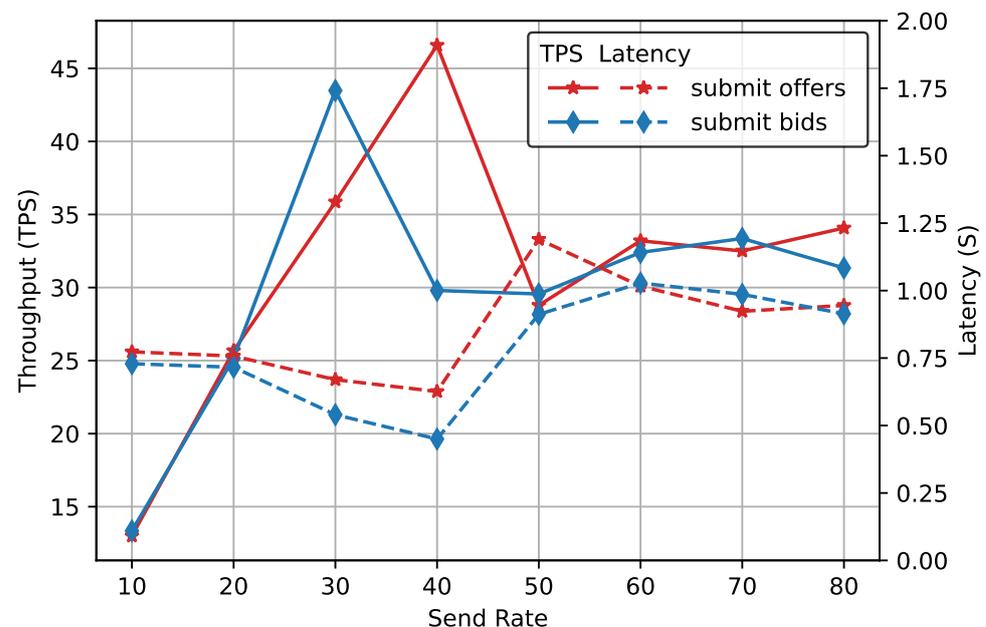
**Figure 6.** Registry contract benchmarking results. Solid lines represent throughput (left y-axis); dashed lines represent latency (right y-axis).

Figure 7 shows the benchmark results of the EnergyToken contract. As we can see, the query throughput dramatically decreases as the transaction send rate increases from 10 TPS to 80 TPS. The blockchain has a relatively high throughput of over 250 TPS under a low query request rate of 10 TPS. When the query request rate increases to an extent, e.g., 80 TPS, network congestion happens, causing a significant performance drop with a throughput of 50 TPS. However, the query latency always maintains a negligible level. The transfer throughput increases to around 48 TPS when the send rate reaches 40 TPS then fluctuates between 30 TPS and 50 TPS. The transfer latency is relatively low and steady, around 0.7 s. The observed experimental results indicate that up to 50 users can send transfer transactions to pay for their energy simultaneously and will obtain a response from the blockchain system within one second.



**Figure 7.** EnergyToken contract benchmarking results. Solid lines represent throughput (left y-axis); dashed lines represent latency (right y-axis).

Figure 8 shows the benchmark results of the PoolMarket contract. As can be seen, both transactions have similar performance trends. The throughput increases linearly from 10 TPS to the maximum of 43 TPS for submitBid and 46 TPS for submitOffer when the send rates reach 30 TPS and 40 TPS, respectively. Then, the throughput drops significantly, followed by a slow and slight increase when the send rate increases to 80 TPS. Accordingly, the latency of both transactions drops from 0.75 s to 0.5 and 0.6 s for submitBid and submitOffer, respectively, at the send rate of 40 TPS. Then, it increases to around one second as the send rate reaches 80 TPS.



**Figure 8.** PoolMarket contract benchmarking results. Solid lines represent throughput (left y-axis); dashed lines represent latency (right y-axis).

The observed experimental results indicate that the blockchain system can process up to 43 submitBid or 46 submitOffer transactions per second. In other words, if 40 users submit their offers or bids simultaneously, they will obtain a successful subsecond response from the blockchain system. In the Alberta electricity wholesale market with a maximum of 245 offer updates in the first daily hour, it only takes up to 6 s for the proposed BPET system to process all offer submission transactions.

## 7. Discussion

### 7.1. Scalability

There are at least three potential methods to improve the performance of the BPET system. First, validators add more resources, e.g., CPU and RAM, to their blockchain nodes and increase the network speed. This method works from the perspective of the vertical scalability [39] of the Hyperledger Besu network. However, adding more resources to nodes can only increase the scalability to a very limited level. Second, the smart contract developer optimizes the registration and energy trading operations to improve the on-chain execution efficiency. This step is based on the fact that the computation complexity in a contract has a significant influence on blockchain performance. For example, under the same network configuration, the throughput (up to 400 TPS) of “transfer” in the Simple contract [39] significantly outperforms “transfer” in the EnergyToken smart contract (up to 48 TPS). Third, smart contracts offload as many on-chain operations (e.g., system marginal price calculation) as possible to off-chain computation, such as Chainlink Functions with a Decentralized Oracle Network (DON) [40]. It is worth noting that the BPET system supports any EVM-compatible blockchain, along with other popular consensus algorithms, such as PoW and PoS. While these consensus mechanisms enhance decentralization, they

also diminish the overall performance and scalability of the blockchain infrastructure. Thereby, more performant and decentralized consensus mechanisms are worth further investigation in the context of the BPET system.

### 7.2. Security

In the BPET system, all validators are entities with their identities verified. A malicious validator voting on false transactions will be detected and voted out of the validators' quorum by all other honest validators through the QBFT consensus. Moreover, the immediate finality property of QBFT eliminates threats of long-range attacks aiming to revise the transaction history of the ledger and selfish mining attacks in which the attacker holds discovered blocks privately and then attempts to fork a private chain.

To eliminate smart contract security vulnerabilities, all BPET smart contracts were fully tested using the Mocha test framework. We also used industrial best practices, including Openzeppelin's [41] secure ERC20 [35] token, access control, and smart contract audit tools (e.g., Slither [42], Securify [43], and Mythril [44]) to ensure the security of BPET smart contracts. However, these security solutions primarily address vulnerabilities in decentralized finance, such as reentrancy, frontrunning, and access control. In contrast, security for energy trading smart contracts heavily relies on self-audit. Therefore, there is significant potential for future research to develop automated solutions for detecting and remedying energy domain-specific smart contract vulnerabilities [45].

### 7.3. Rollups

Rollups are a popular development of the Layer 2 scaling strategy, which performs calculations off-chain, rolls many transactions up into a single batch, and sends it to the Ethereum Mainnet or its compatible chains (Layer 1) in a single action. They significantly decrease associated transaction processing times and gas fees and increase the system throughput simultaneously. Currently, there are two major types of rollups [46] used for scaling Ethereum: Zero-Knowledge Rollups (ZK Rollups), e.g., zkSync, StarkNet, Scroll, Polygon zkEVM, and Taiko, etc., and Optimistic Rollups, e.g., Arbitrum, Optimism, Metis Andromeda, and Boba Network, etc.

The main difference between ZK and Optimistic rollups lies in how to finalize this batch of transactions. In ZK rollups, the Ethereum network verifies the correctness of the batch of transactions through cryptographic validity proofs (commonly called zero-knowledge proofs). Optimistic rollups, on the other hand, assume that off-chain transactions are valid unless proven otherwise. Hence, they rely on fraud proofs, a challenge to the submitted state to Ethereum.

Using the proposed BPET framework, we built another decentralized electricity exchange system called zkDELX [47], which implemented a bilateral market on Polygon, Scroll, and Taiko zkEVMS. This market enables electric vehicle users to share their home chargers and exchange electricity in a peer-to-peer way to resolve the charging station shortage issue through the shared economy business model.

## 8. Conclusions

Blockchain technology has received substantial attention for its ability to address security, privacy, and reliability challenges inherent to decentralized energy trading solutions. However, the diversity of the energy markets and the absence of an extensible framework impede the widespread adoption of blockchain-based P2P-ET solutions globally. In this study, we introduce a novel unified and extensible blockchain-based P2P-ET framework, BPET, to bridge this gap. This framework adopts the performant private Hyperledger Besu blockchain and microservices as its infrastructures and enables developers to swiftly build a localized P2P-ET system on top of it. We designed and implemented a prototype of BPET by developing modular smart contracts, deploying them onto the blockchain network, and constructing a decentralized application. To validate the proposed solution, we conducted a case study using real AESO energy data and evaluated the performance of

smart contracts using the Hyperledger Caliper benchmark tool. The experimental results demonstrate the system's capability of handling energy trading transactions effectively and efficiently within the Alberta electricity wholesale market. For future research topics, it is of interest to explore the integration of automated market maker mechanisms to enhance market liquidity and efficiency. It is also interesting to develop and incorporate more energy markets as practical examples to enhance the usability of the BPET framework. Additionally, our interest is to investigate the feasibility of incorporating cutting-edge zero-knowledge machine learning technologies into P2P-ET to enhance the intelligence and privacy of blockchain-based energy systems.

**Author Contributions:** Conceptualization, C.F.; methodology, C.F., H.K. and P.M.; software, C.F.; validation, C.F.; formal analysis, C.F. and H.K.; investigation, C.F.; resources, P.M.; data curation, P.M.; writing original draft preparation, C.F.; writing, review, and editing, C.F., H.K. and P.M.; visualization, C.F.; supervision, H.K. and P.M.; project administration, P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Huawei-ECE Research Initiative at the University of Alberta.

**Data Availability Statement:** All original data used in this research are from the Alberta Electric System Operator. The real-time and historical provincial electric wholesale market data are publicly accessible at <http://ets.aeso.ca/> (accessed on 5 May 2024).

**Acknowledgments:** Digital Research Alliance of Canada (alliancecan.ca) and Cybera (cybera.ca) partially supported this research work through their cloud computing resources.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

DERs	distributed energy resources
P2P-ET	peer-to-peer energy trading
BPET	blockchain-based peer-to-peer energy trading
PoW	proof of work
PoS	proof of stake
PoA	proof of authority
QBFT	quorum Byzantine fault tolerance
IBFT	Istanbul Byzantine fault tolerance
DSO	distribution system operators
TSO	transmission system operators
MO	market operators
MCP	market clearance price
SMP	system marginal price
SMO	system marginal offer
AIL	Alberta internal load
AESO	Alberta Electricity System Operator
AEMO	Australian Energy Market Operator
VET	volume of energy to be traded
ESA	energy supply aggregate
EDA	energy demand aggregate
LAB	load amount of bid
ETK	energy token
TPS	transactions per second
ZK	zero-knowledge

## References

- Zhang, C.; Wu, J.; Zhou, Y.; Cheng, M.; Long, C. Peer-to-Peer energy trading in a Microgrid. *Appl. Energy* **2018**, *220*, 1–12. [CrossRef]
- Soto, E.A.; Bosman, L.B.; Wollega, E.; Leon-Salas, W.D. Peer-to-peer energy trading: A review of the literature. *Appl. Energy* **2021**, *283*, 116268. [CrossRef]
- Mollah, M.B.; Zhao, J.; Niyato, D.; Lam, K.Y.; Zhang, X.; Ghias, A.M.; Koh, L.H.; Yang, L. Blockchain for future smart grid: A comprehensive survey. *IEEE Internet Things J.* **2020**, *8*, 18–43. [CrossRef]
- Hasan, M.K.; Alkhalifah, A.; Islam, S.; Babiker, N.B.; Habib, A.A.; Aman, A.H.M.; Hossain, M.A. Blockchain technology on smart grid, energy trading, and big data: Security issues, challenges, and recommendations. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9065768. [CrossRef]
- Andreoulaki, I.; Papapostolou, A.; Marinakis, V. Evaluating the Barriers to Blockchain Adoption in the Energy Sector: A Multicriteria Approach Using the Analytical Hierarchy Process for Group Decision Making. *Energies* **2024**, *17*, 1278. [CrossRef]
- Koukaras, P.; Afentoulis, K.D.; Gkaidatzis, P.A.; Mystakidis, A.; Ioannidis, D.; Vagropoulos, S.I.; Tjortjis, C. Integrating Blockchain in Smart Grids for Enhanced Demand Response: Challenges, Strategies, and Future Directions. *Energies* **2024**, *17*, 1007. [CrossRef]
- Vionis, P.; Kotsilieris, T. The Potential of Blockchain Technology and Smart Contracts in the Energy Sector: A Review. *Appl. Sci.* **2023**, *14*, 253. [CrossRef]
- Aitzhan, N.Z.; Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852. [CrossRef]
- Gai, K.; Wu, Y.; Zhu, L.; Qiu, M.; Shen, M. Privacy-preserving energy trading using consortium blockchain in smart grid. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3548–3558. [CrossRef]
- Abdella, J.; Tari, Z.; Anwar, A.; Mahmood, A.; Han, F. An Architecture and Performance Evaluation of Blockchain-based Peer-to-Peer Energy Trading. *IEEE Trans. Smart Grid* **2021**, *12*, 3364–3378. [CrossRef]
- Lohachab, A.; Garg, S.; Kang, B.H.; Amin, M.B. Performance evaluation of Hyperledger Fabric-enabled framework for pervasive peer-to-peer energy trading in smart Cyber-Physical Systems. *Future Gener. Comput. Syst.* **2021**, *118*, 392–416. [CrossRef]
- Hahn, A.; Singh, R.; Liu, C.C.; Chen, S. Smart contract-based campus demonstration of decentralized transactive energy auctions. In Proceedings of the 2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 23–26 April 2017; pp. 1–5.
- Faizan, M.; Brenner, T.; Foerster, F.; Wittwer, C.; Koch, B. Decentralized bottom-up energy trading using Ethereum as a platform. *J. Energy Mark.* **2019**, *12*, 19–48. [CrossRef]
- 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]
- AlSkaif, T.; Crespo-Vazquez, J.L.; Sekuloski, M.; van Leeuwen, G.; Catalão, J.P. Blockchain-based fully peer-to-peer energy trading strategies for residential energy systems. *IEEE Trans. Ind. Inform.* **2021**, *18*, 231–241. [CrossRef]
- Yang, J.; Paudel, A.; Gooi, H.B.; Nguyen, H.D. A Proof-of-Stake public blockchain based pricing scheme for peer-to-peer energy trading. *Appl. Energy* **2021**, *298*, 117154. [CrossRef]
- Wang, T.; Guo, J.; Ai, S.; Cao, J. RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration. *Appl. Energy* **2021**, *295*, 117056. [CrossRef]
- Ali, F.S.; Bouachir, O.; Özkasap, Ö.; Aloqaily, M. SynergyChain: Blockchain-assisted adaptive cyber-physical P2P energy trading. *IEEE Trans. Ind. Inform.* **2020**, *17*, 5769–5778. [CrossRef]
- Baig, M.J.A.; Iqbal, M.T.; Jamil, M.; Khan, J. A low-cost, open-source peer-to-peer energy trading system for a remote community using the internet-of-things, blockchain, and hypertext transfer protocol. *Energies* **2022**, *15*, 4862. [CrossRef]
- Mehdinejad, M.; Shayanfar, H.; Mohammadi-Ivatloo, B. Decentralized blockchain-based peer-to-peer energy-backed token trading for active prosumers. *Energy* **2022**, *244*, 122713. [CrossRef]
- Condon, F.; Franco, P.; Martínez, J.M.; Eltamaly, A.M.; Kim, Y.C.; Ahmed, M.A. EnergyAuction: IoT-Blockchain Architecture for Local Peer-to-Peer Energy Trading in a Microgrid. *Sustainability* **2023**, *15*, 13203. [CrossRef]
- Veerasingh, V.; Hu, Z.; Qiu, H.; Murshid, S.; Gooi, H.B.; Nguyen, H.D. Blockchain-enabled peer-to-peer energy trading and resilient control of microgrids. *Appl. Energy* **2024**, *353*, 122107. [CrossRef]
- Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3690–3700. [CrossRef]
- Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3154–3164. [CrossRef]
- Chaudhary, R.; Jindal, A.; Auja, G.S.; Aggarwal, S.; Kumar, N.; Choo, K.K.R. BEST: Blockchain-based secure energy trading in SDN-enabled intelligent transportation system. *Comput. Secur.* **2019**, *85*, 288–299. [CrossRef]
- Sheikh, A.; Kamuni, V.; Urooj, A.; Wagh, S.; Singh, N.; Patel, D. Secured energy trading using byzantine-based blockchain consensus. *IEEE Access* **2019**, *8*, 8554–8571. [CrossRef]
- Sadiq, A.; Javed, M.U.; Khalid, R.; Almogren, A.; Shafiq, M.; Javaid, N. Blockchain based data and energy trading in internet of electric vehicles. *IEEE Access* **2020**, *9*, 7000–7020. [CrossRef]
- Mengelkamp, E.; Gärttner, J.; Rock, K.; Kessler, S.; Orsini, L.; Weinhardt, C. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Appl. Energy* **2018**, *210*, 870–880. [CrossRef]

29. Long, C.; Wu, J.; Zhang, C.; Thomas, L.; Cheng, M.; Jenkins, N. Peer-to-peer energy trading in a community microgrid. In Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5.
30. Wilkins, D.J.; Chitchyan, R.; Levine, M. Peer-to-peer energy markets: Understanding the values of collective and community trading. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–14.
31. Sousa, T.; Soares, T.; Pinson, P.; Moret, F.; Baroche, T.; Sorin, E. Peer-to-peer and community-based markets: A comprehensive review. *Renew. Sustain. Energy Rev.* **2019**, *104*, 367–378. [CrossRef]
32. Sensfuß, F.; Ragwitz, M.; Genoese, M. The merit-order effect: A detailed analysis of the price effect of renewable electricity generation on spot market prices in Germany. *Energy Policy* **2008**, *36*, 3086–3094. [CrossRef]
33. AEMO. About AEMO. 2024. Available online: <https://aemo.com.au/about> (accessed on 12 April 2024).
34. AESO. About the AESO. 2024. Available online: <https://www.aeso.ca/aeso/about-the-aeso/> (accessed on 12 April 2024).
35. Bauer, D.P. ERC-20: Fungible Tokens. In *Getting Started with Ethereum: A Step-by-Step Guide to Becoming a Blockchain Developer*; Springer: Berkeley, CA, USA, 2022; pp. 17–48.
36. Fan, C. BPET-Contracts. 2024. Available online: <https://github.com/CaixiangFan/bpet/tree/main> (accessed on 12 April 2024).
37. Utilities Consumer Advocate. Retailers and Distributors. 2024. Available online: <https://ucahelps.alberta.ca/retailers.aspx> (accessed on 12 April 2024).
38. DRAC. Cloud Resources. 2024. Available online: [https://docs.computecanada.ca/wiki/Cloud\\_resources](https://docs.computecanada.ca/wiki/Cloud_resources) (accessed on 12 April 2024).
39. Fan, C.; Lin, C.; Khazaei, H.; Musilek, P. Performance Analysis of Hyperledger Besu in Private Blockchain. In Proceedings of the 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), Newark, CA, USA, 15–18 August 2022; pp. 64–73.
40. Chainlink. What Is Chainlink Functions? 2024. Available online: <https://docs.chain.link/chainlink-functions> (accessed on 12 April 2024).
41. OpenZeppelin. The Standard for Secure Blockchain Applications. 2024. Available online: <https://www.openzeppelin.com/> (accessed on 12 April 2024).
42. Feist, J.; Grieco, G.; Groce, A. Slither: A static analysis framework for smart contracts. In Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Montreal, QC, Canada, 27 May 2019; pp. 8–15.
43. Tsankov, P.; Dan, A.; Drachler-Cohen, D.; Gervais, A.; Buenzli, F.; Vechev, M. Securify: Practical security analysis of smart contracts. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 67–82.
44. Sharma, N.; Sharma, S. A survey of Mythril, a smart contract security analysis tool for EVM bytecode. *Indian J. Nat. Sci.* **2022**, *13*, 75.
45. Lashkari, B.; Musilek, P. Detection and Analysis of Ethereum Energy Smart Contracts. *Appl. Sci.* **2023**, *13*, 6027. [CrossRef]
46. Čapko, D.; Vukmirović, S.; Nedić, N. State of the Art of Zero-Knowledge Proofs in Blockchain. In Proceedings of the 2022 30th Telecommunications Forum (TELFOR), Belgrade, Serbia, 15–16 November 2022; pp. 1–4.
47. Fan, C.; He, T. zkDELX. 2024. Available online: <https://github.com/orgs/zk-DELX/repositories> (accessed on 12 April 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.