Annals of Telecommunications
https://doi.org/10.1007/s12243-023-00973-8

On the performance and scalability of consensus mechanisms in privacy-enabled decentralized renewable energy marketplace

Roman-Valentyn Tkachuk[1] · **Dragos Ilie**[1] · **Remi Robert**[2] · **Victor Kebande**[1] · **Kurt Tutschku**[1]

Received: 30 April 2023 / Accepted: 20 June 2023
© The Author(s) 2023

Abstract

Renewable energy sources were introduced as an alternative to fossil fuel sources to make electricity generation cleaner. However, today's renewable energy markets face a number of limitations, such as inflexible pricing models and inaccurate consumption information. These limitations can be addressed with a decentralized marketplace architecture. Such architecture requires a mechanism to guarantee that all marketplace operations are executed according to predefined rules and regulations. One of the ways to establish such a mechanism is blockchain technology. This work defines a decentralized blockchain-based peer-to-peer (P2P) energy marketplace which addresses actors' privacy and the performance of consensus mechanisms. The defined marketplace utilizes private permissioned Ethereum-based blockchain client Hyperledger Besu (HB) and its smart contracts to automate the P2P trade settlement process. Also, to make the marketplace compliant with energy trade regulations, it includes the regulator actor, which manages the issue and consumption of guarantees of origin and certifies the renewable energy sources used to generate traded electricity. Finally, the proposed marketplace incorporates privacy-preserving features, allowing it to generate private transactions and store them within a designated group of actors. Performance evaluation results of HB-based marketplace with three main consensus mechanisms for private networks, i.e., Clique, IBFT 2.0, and QBFT, demonstrate a lower throughput than another popular private permissioned blockchain platform Hyperledger Fabric (HF). However, the lower throughput is a side effect of the Byzantine Fault Tolerant characteristics of HB's consensus mechanisms, i.e., IBFT 2.0 and QBFT, which provide increased security compared to HF's Crash Fault Tolerant consensus RAFT.

Keywords Renewable energy marketplace · Blockchain technology · Peer-to-peer energy trading · Hyperledger Besu · Data privacy

Abbreviations

BFT	Byzantine fault tolerant
BO	Blockchain organization
BPS	Block period seconds
CFT	Crash fault tolerant
D2018/2001	Directive 2018/2001
DER	Distributed energy resource
DoS	Denial of service
DSL	Domain-specific language
EIP	Ethereum Improvement Proposal
FT	Fungible token
GDPR	General Data Protection Regulation
GO	Guarantee of origin
HB	Hyperledger Besu
HC	Hyperledger caliper
HF	Hyperledger fabric
IAM	Identity and access management
IBFT	Istanbul BFT
kWh	Kilowatt-hours
MI	Marketplace interface
NFT	Non-fungible token

✉ Roman-Valentyn Tkachuk
roman-valentyn.tkachuk@bth.se

Dragos Ilie
dragos.ilie@bth.se

Remi Robert
remi.robert@ericsson.com

Victor Kebande
victor.kebande@bth.se

Kurt Tutschku
kurt.tutschku@bth.se

[1] Department of Computer Science, Blekinge Institute of Technology, Karlskrona, Sweden

[2] Ericsson Research, Stockholm, Sweden

Published online: 02 September 2023

P2P	Peer-to-peer
PG	Privacy group
PoA	Proof of authority
PoS	Proof of stake
PoW	Proof of work
QBFT	Quorum BFT
QoS	Quality of service
RES	Renewable energy source
RPS	Reads per second
SC	Smart contract
TPS	Transactions per second
TTP	Trusted third party

1 Introduction

Energy distribution systems play a vital role in the modern world. The dependency on electricity supply transcends every aspect of a society's operation, making it a necessity. However, the electricity production conducted by power plants that work on fossil fuels results in atmosphere carbonization. In order to make electricity generation cleaner, renewable energy sources (RESs), e.g., solar panels, were introduced as an alternative to fossil fuel ones. Consequently, the introduction of RES opened opportunities for electricity prosumers, i.e., producers/consumers, to become a part of the grid as a distributed energy resource (DER) [1]. This allows prosumers to not only consume energy as a conventional node but also to produce and output it to the energy grid [2]. Further, prosumers can also trade the produced electricity through the energy marketplace, which incentivizes the installation of RES and the production of green electricity. However, today's energy markets face a number of challenges when it comes to management and operation.

The first is the *inflexible pricing model* of today's marketplaces. In such a model, the prosumer is limited to selling the generated electricity to a single buyer without any other options, e.g., it is sold to an energy provider who owns the grid to which the prosumer's RES is connected. In addition, the generated electricity is sold at a price set by the buyer through a governmental body, e.g., a country's energy agency regulates the margins for the RES-produced electricity trade and does not provide any room for negotiation. This creates a number of limitations for prosumers within an energy marketplace. It limits the volume of consumers that the prosumer can reach to sell their RES-produced electricity. Further, the seller cannot reach consumers belonging to a different electricity provider. Finally, this challenge results in a value distribution imbalance, where the prosumer side is losing a part of electricity sale profits due to price inflexibility [3].

The second challenge is *inaccurate green consumption information*, i.e., buyers receive unreliable information about the sources of the electricity they consume. Ultimately, the consumers are ready to pay higher electricity prices for RES-produced electricity to support the decarbonization of the atmosphere. Due to the inflexibility of the energy grid and inaccurate national regulatory frameworks, consumers frequently end up using electricity generated by fossil fuel sources while paying for RES-generated energy. This results in the devaluation of RES-produced electricity as prosumers do not see the benefit in buying it while being supplied with fossil fuel produced energy. Nowadays, the information about RES-produced electricity is recorded in the *guarantee of origin* (GO). GO is proof to the buyer that the electricity at a given quantity was produced by the RES [4]. Typically, the GO is issued by the governmental *regulator*, who certifies the prosumer-owned RES and an associated electricity generation metering device. However, due to the inflexibility of energy distribution systems, e.g., unavailability of RES in close proximity to consumers, they still end up using the electricity which was produced by fossil fuel energy sources while having the GO [5].

These limitations can be alleviated by introducing the *peer-to-peer* (P2P) *electricity trading*, which is an automated sale process for renewable energy between market participants using a contract with pre-determined conditions [4]. A P2P energy trade settlement allows prosumers to trade electricity directly with each other, enabling them to control when, where, and for what price the electricity is bought or sold. The ultimate goal of P2P energy trading is to create an incentive for the widespread adoption of RESs, resulting in the decarbonization of the energy distribution systems [6].

Today's marketplaces are built as centralized systems. Thus, a *trusted third-party* (TTP) (typically a prosumer's energy provider) has to be present to guarantee that the predetermined conditions of a P2P energy trading contract are followed. However, trust issues are raised, when it comes to scaling the marketplace to more than one energy provider. Energy providers want to keep their operations private to maintain a competitive advantage in the electricity market. This requires the introduction of an external TTP that can be trusted by all energy providers within the marketplace [7], i.e., allowing individuals belonging to different energy providers to trade with each other. To remediate these limitations, a *decentralized marketplace architecture* can be used to distribute control over the marketplace operations to multiple energy providers. However, all organizations require an efficient and robust consensus-reaching mechanism that provides guarantees that P2P trade settlement conditions are followed while maintaining actors' data privacy. Such capabilities can be provided by blockchain technology [8]. Blockchain provides marketplace participants with distributed storage, i.e., the ledger, and brings such benefits as

provenance, accountability, and privacy to all data processed in a system. It also acts as a consensus-reaching platform, allowing initially non-trusting energy providers and prosumers to establish a trusted relationship and conduct P2P trade settlements without needing a single TTP acting as a middleman [9].

Based on the challenges discussed above, the main contributions of this study can be summarized as follows. This study defines a decentralized blockchain-based P2P energy marketplace that utilizes *Hyperledger Besu* (HB) [10] as the blockchain platform. The proposed marketplace utilizes HB's *smart contracts* (SCs) to automate P2P energy trade settlement and issue and consume GOs. To make the marketplace compliant with energy trade regulations, it incorporates the *regulator* actor, which manages the issue and consumption of GO and certifies the RES used to generate traded electricity. Further, the marketplace utilizes Tessera private transaction manager to ensure actor data privacy. The following methodology was used to define the proposed marketplace. First, with advice from an energy provider, we define a set of regulatory and operational requirements. Further, we define the marketplace's architecture and detail its implementation. Next, we present the performance evaluation with the SC tailored for P2P energy trading. We investigate in-depth the performance of the main Proof of Authority (PoA) consensus mechanisms supported by HB, i.e., QBFT, IBFT 2.0, and Clique. Finally, we summarize observations on the mechanisms that lead to secure consensus while preserving actors' data privacy. This paper is an extension of [11], and provides extended discussions on system architecture, implementation, performance evaluation results, and a summary.

The remainder of the paper is structured as follows. Section 2 describes the actors for the proposed marketplace and details its blockchain platform. Section 3 details the marketplace implementation, data structure, and smart contract definition. Section 4 details the performance evaluation process and results. Section 5 describes the observations from marketplace implementation. Section 6 describes related work on energy marketplaces and their performance evaluation. Finally, Sect. 7 summarizes the investigation of the proposed marketplace and provides an outlook.

2 Blockchain-based energy marketplace

The energy marketplace is subject to several regulatory constraints, which must be met to satisfy current P2P energy trade regulations, i.e., GOs and an automated trade contract. Thus, the proposed marketplace requirements are aligned with regulations described in Directive 2018/2001 (D2018/2001) of the European Parliament [4] regarding the issuing, trading, and consumption of GOs. To align with D2018/2001, we introduce a *regulator* role in the proposed marketplace. The regulator is an actor that manages the issue and consumption of GOs, which are required to execute a trade settlement contract. Further, the regulator certifies the RES used to generate traded electricity. To the best of our knowledge, none of the other energy marketplace studies considers such governmental regulatory requirements in conjunction with actors' data privacy. Finally, the marketplace actors and requirements were defined in collaboration with the authors' local energy provider, which has DERs as a part of their grid infrastructure.

2.1 Marketplace actors and requirements

The actors and requirements were defined in collaboration with the authors' local energy provider which has DERs as part of its grid infrastructure. Further, the requirements were defined in compliance with the regulations described in D2018/2001. Energy marketplace actors and their respective places in the grid infrastructure are depicted in Fig. 1.

The *prosumer* represents a DER in an energy grid with an installed RES. The prosumer's main interest in becoming a part of the marketplace is to control the conditions of energy trade settlement, e.g., to sell electricity at a better price. Further, prosumers want to get GO for the electricity they produce within the marketplace's automated system.

The *energy provider* is an actor that manages the energy grid to which the prosumer is connected. As a local central point in the energy distribution scheme, the energy provider collects data on electricity consumption fluctuations to optimize distribution and conduct an accounting of the electricity and money flows in its network. Further, energy providers want to expand their DER infrastructure to meet customer demand for RES-generated energy delivery.

The *regulator* is the representative of governmental authority who manages the issue and consumption of GOs. The GO acts as proof that the electricity was generated with RES and must be presented during the trade transaction by the prosumer-seller. Further, the regulator is the entity that certifies prosumers' RES and ensures the correct mapping between the generated and marketplace-traded electricity.

2.1.1 Functional requirements

To enable renewable energy trade settlement, a set of operations must be defined. The *functional requirements* [12] define the operations which can be executed by the prosumers, energy providers, and regulators within the marketplace.

Electricity operations The prosumers must have the ability to *manage the generated electricity*, which is represented by virtual kilowatt-hours (kWh) on the level of the marketplace trade operations. Within the marketplace, generated

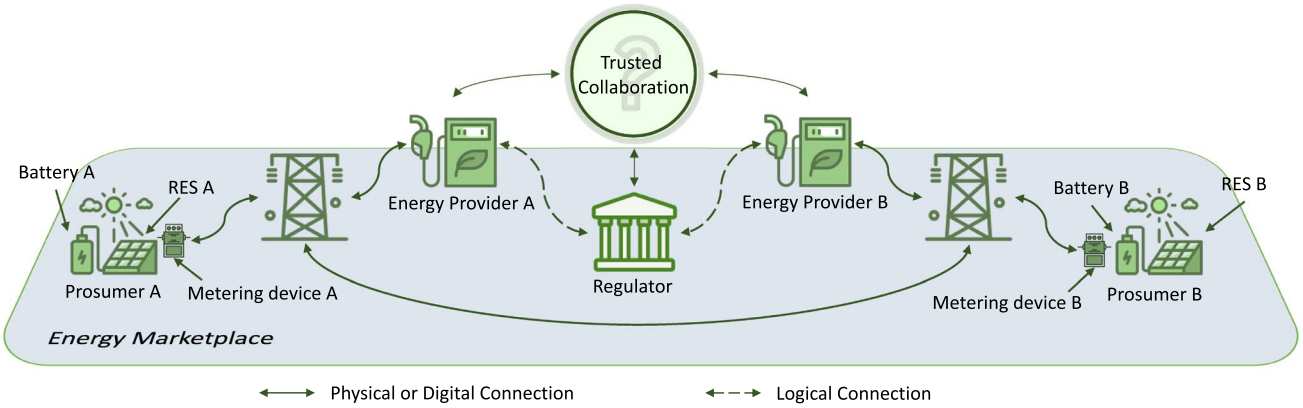

Fig. 1 Energy marketplace actors (the "Trusted Collaboration" component signifies the need to enforce rules that all actors of the marketplace follow)

electricity acts as a *fungible token* (FT) [13]. First, the prosumer *registers generated electricity* by adding a number of virtual kWh to their marketplace account. This happens automatically via a metering device connected to the prosumer's RES. Further, the prosumer should be able to *trade the generated electricity*, i.e., sell/buy electricity at a given price. The associated GO is consumed when the electricity is sold, making it impossible to further sell it to another prosumer.

Ordering system operations Prosumers propose the energy trade through a *marketplace ordering system*. First, the marketplace should enable the creation of *offer to sell electricity* of a given quantity at a given price. Further, the marketplace should enable the creation of an *offer to buy electricity* of a given quantity at a given price.

GO operations The regulator issues the GO on the electricity generated by a specific RES. The GO acts as a *non-fungible token* (NFT) [14]. When the electricity is generated by a RES, the regulator should be able to *issue the GO* on the name of the prosumer, which is presented when the energy is sold. Further, when the electricity is sold, the GO should be *consumed*, to disable the double selling of energy.

2.1.2 Non-functional requirements

The *non-functional requirements* [15] define the global constraints which affect the marketplace system's reliability and data assurance.

Data correctness The marketplace must ensure that the virtual kWh must only be *issued following the actual generation of electricity*. When an order is executed, the marketplace has to *make sure that appropriate resources*, i.e., virtual kWh and currency, *are available* for both seller and buyer. In addition, the *energy trade must be executed to a set of conditions that were previously approved by marketplace actors*. Further, the *GO must only be issued following the generation of electricity from renewable sources*, e.g., hydro, wind, or solar [16]. Finally, *it must be impossible to sell consumed GO*.

Data privacy To ensure data privacy [17], *all transactions from a prosumer, including generation, selling, and purchase, should not be disclosed to other prosumers*. Further, the *details of P2P energy trade should be disclosed only to the prosumers, their respective energy providers, and the regulator*. Finally, *prosumer energy generation information has to be visible for the regulator* to ensure correct mapping between virtual kWh and actual generated electricity.

2.2 Blockchain platform

Blockchain technology [8] can be used to provide the technical building block allowing for meeting the requirements defined in Sect. 1. Blockchain provides marketplace participants with distributed storage, i.e., the ledger, and brings accountability and provenance to all data processed in a network. With a consensus mechanism, blockchain allows initially non-trusting energy providers, regulators, and prosumers to establish a trusted relationship [18]. Thus, blockchain technology removes the need for a single TTP accepted by all marketplace actors.

To enable marketplace actors to conduct P2P trade settlement, a blockchain platform has to be chosen such that it meets the identified requirements. Considering the privacy requirements, the proposed marketplace utilizes a private permissioned blockchain platform. Permissioned blockchain network has an identity and access management (IAM) [19] mechanism that defines a set of entities, i.e., collaborating organizations and users, which are allowed

to access the network. Further, permissioned blockchain requires that after entering the network, the entity has to be authorized to execute new transactions and add them to the global ledger. Finally, private blockchains enable data privacy and better address the demands of the business use cases [20]. Here, the data privacy mechanism is defined as the ability to keep blockchain transactions private for a certain group of participants. *Hyperledger Besu* (HB) [10] is representative of private permissioned blockchain platform. It is an open-source Ethereum [21] client that PegaSys[1] first developed and later handed over to the Hyperledger Foundation.[2] From the beginning, the Ethereum blockchain was designed as a public permissionless platform, i.e., opened for everyone to join and generate transactions. HB can be considered an adaptation of the original public Ethereum blockchain to the private context. Here, HB implements the *Enterprise Ethereum Alliance Protocol* to enable such functionality as private transactions, IAM, and permissioning. In the HB network, the *validator* nodes order, execute, and verify transactions in the blockchain network. However, validator nodes cannot be used to initiate transactions. All transactions in the HB network are initiated by *user accounts*, representing a public and private key pair that can be generated off-chain. The smart contract (SC) defines functions that a user account can call to operate on the data in the ledger. First, the SC has to be installed in the blockchain network. Once installed, it serves as a predefined trade settlement contract where fixed, agreed-upon rules are enforced during every execution.

2.3 Identity and access management

The IAM in HB can be implemented using *local* and *on-chain* permissioning. The local permissioning is defined in a *permissions configuration file* and can be specified on each individual blockchain node. This permissioning type does not require consensus from the rest of the network. Local permissioning allows the specification of the list of valid nodes to which the validator can connect. In addition, it allows specifying the user accounts that can use the validator to execute transactions. In contrast, the on-chain permissioning is defined by the *permissioning management SC* and requires consensus of all nodes in the network. The SC acts as a program within a marketplace that collaborating organizations, i.e., admins, install in the HB network to govern the IAM. Through the SC, admins can specify a list of nodes authorized to be a part of the network and perform consensus mechanism, i.e., order, execute, and verify transactions. Further, through the SC, admins can specify the user accounts that can initiate transactions in the network.

[1] A team of engineers within ConsenSys company.

[2] https://www.hyperledger.org/

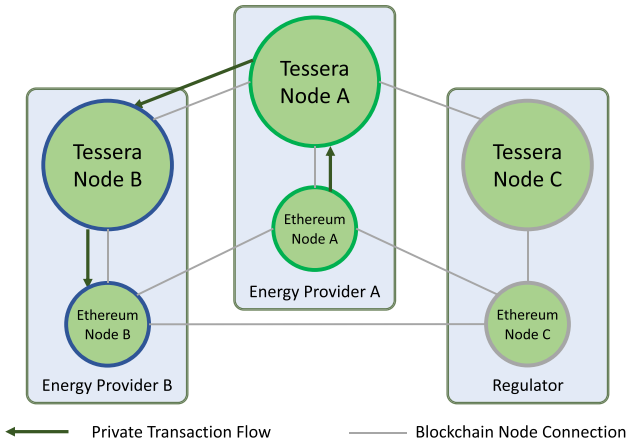

Fig. 2 Implemented energy marketplace

The advantage of on-chain permissioning is the ability to collectively govern access to the HB network and track the changes to access lists of full nodes and user accounts.

2.4 Data privacy

In HB, private data is stored in transactions that are disclosed only to a subset of network participants (further referred to as privacy group (PG)), while the rest of the network does not have access to the contents. Further, the rest of the network does not know the list of nodes that belong to PG. The private transactions in HB are handled by the *Tessera* private transaction manager. Each organization in HB must have a Tessera node to participate in private transactions. When a new private transaction is generated, it is passed from the Ethereum node to the Tessera node associated with it. Further, the Tessera node encrypts the transaction and distributes it to the PG. Recipient Tessera Nodes from the PG decrypt the transaction and pass it to their Ethereum Nodes. The rest of the nodes outside of the PG receive the record confirming that the private transaction was executed. Such a record consists of a hash of the encrypted transaction data and a privacy marker, i.e., indicator that the transaction is private. Further, this record is written into the global ledger.

The HB private transaction flow is shown in Fig. 2. *Energy Provider A* and *Energy Provider B* are in the same PG. When *Ethereum Node A* generates the private transaction, it is passed to the *Tessera Node A* for encryption. Further, *Tessera Node A* transmits the encrypted transaction to *Tessera Node B*, where it is decrypted and written to the private storage of *Ethereum Node B*. Finally, the *Regulator's Ethereum Node C* receives the record with the encrypted transaction data and a privacy marker which is written to the global ledger.

In the public Ethereum network, *gas* is the unit of measurement for the number of computations needed to execute a transaction. The user is required to pay a certain amount of *gas*, i.e., depending on complexity, in order to execute a transition. In contrast, privacy-enabled HB Ethereum networks allow disabling gas spending to execute both ordinary and private transactions. This requires a certain level of trust among the blockchain network transacting nodes, i.e., that none of the participants will act maliciously and perform a denial of service (DoS) attack by flooding the network with transactions. Thus, privacy-enabled networks must have off-chain trust-enabling mechanisms, including SC deployment recommendations and legal consequences for malicious activity.

2.5 Smart contract

The smart contract (SC) in HB represents a concrete entity with functions that a user account can call. An SC cannot be triggered by any other internal HB network event or entity. Further, SCs are isolated in terms of storage, i.e., each SC has its namespace and operates on the records saved there. However, one SC can invoke the functions of other SCs.

Initially, an SC is written in a high-level programming language, e.g., Solidity[3] or Vyper.[4] These are domain-specific languages (DSLs), i.e., defined to work specifically with Ethereum SCs. One prominent Ethereum DSL is Solidity, which is influenced by JavaScript in terms of syntax and structure. There are various open-source Solidity libraries that can be reused and adapted to a specific use case. Solidity enables the development of complex SCs, i.e., the syntax and code constructions facilitate the implementation of complex function routines. One disadvantage is this may lead to an introduction of security vulnerabilities since complex routines may not behave as expected after the SC compilation [22], i.e., due to the inexperience of the developer. Vyper is a DSL designed specifically for Ethereum to improve the auditability and security of SCs. In addition, Vyper DSL has a simpler syntax in comparison to Solidity. An SC code written in Vyper has built-in controls which prevent the introduction of security vulnerabilities. Further, Vyper SC can be more comprehensively reviewed by all collaborating parties relying on built-in security controls. After the SC is finalized in DSL, it is compiled into the *runtime bytecode*, i.e., a state in which the SC is saved on the ledger. Further, SCs are executed in the Ethereum virtual machine (EVM), i.e., an executable environment that is deployed locally for each validator.

[3] https://soliditylang.org/

[4] https://docs.vyperlang.org/

Table 1 Besu consensus mechanisms comparison

Property	Ethash	PoS	Clique	IBFT 2.0	QBFT
Type	PoW	PoS	PoA	PoA	PoA
Finality	No	No	No	Yes	Yes
Quorum	1/2	1/2	NA	2/3	2/3
BFT	Yes	Yes	No	Yes	Yes
Liveness	1/2	1/2	1/2	1/3	1/3
Network	Public	Public	Private	Private	Private

2.6 Consensus mechanisms

The consensus mechanism defines an algorithm by which all nodes in the network can agree on the validity of transaction order in the block. While proof of work (PoW) [8] worked in a public blockchain, it was unsuitable for private deployment, i.e., low transaction throughput and high energy consumption to mine new blocks. Hence, a new approach was followed in private Ethereum called proof of authority (PoA). The blocks in PoA consensus mechanisms are not mined but signed by the designated pool of validators, i.e., avoid wasting energy by delegating block creation to the trusted nodes.

Within the available consensus mechanisms, some are identified as Byzantine fault tolerant (BFT) and/or crash fault tolerant (CFT) [23]. CFT consensus mechanisms are protected only from node failure, i.e., if less than 50% of the nodes fail, the network can operate successfully. BFT consensus provides the same level of protection as CFT and in addition can operate in the presence of adversaries, e.g., nodes that manipulate transactions and try to disrupt the blockchain network operation. However, there are limitations to the BFT consensus mechanisms in terms of the number of adversaries, i.e., consensus is jeopardized if more than one-third of the nodes collude. In practice, when the blockchain user account initiates the transaction, it must wait until the moment the $2\,m+1$ responses are received, where m is the maximum number of allowed failed or malicious nodes. When $2\,m+1$ responses are successfully received, the consensus is achieved and the state of the network is updated. The improved security of BFT consensus mechanisms may come at the cost of decreased performance compared to CFT ones.

The consensus mechanisms supported by the HB are PoW (Ethash), Proof of Stake (PoS), and PoA (Clique, IBFT 2.0, and QBFT). A brief summary of all HB consensus mechanisms characteristics is listed in Table 1. This study concentrates on PoA consensus mechanisms used in private HB networks. When comparing consensus mechanisms, such characteristics as *immediate finality*, *quorum*, *liveness*, and *throughput* have to be considered. Immediate finality refers to the ability to avoid forks, i.e., alternative blockchains or

chain reorganizations. Quorum refers to the minimum number of validator nodes in the blockchain network. Liveness refers to how many failed validators it can sustain and continue normal operation. Throughput refers to the maximum number of write or read transactions, c.f., Sect. 4. The characteristics of each investigated consensus mechanism are discussed next.

Clique is a PoA consensus algorithm that was first proposed in the Ethereum Improvement Proposal (EIP) [24]. In Clique, a designated node pool of trusted *signers*, i.e., validators, creates and adds a new block to the ledger. Further, the existing pool of signers in Clique can vote to include a new or exclude an existing signer. The list of trusted signers is saved on the ledger to ensure that the correct pool is always accessible to every signer in the network. The block creation process is called *sealing*, where signers create new blocks at a fixed time interval defined in *block period seconds*. When block period seconds time is up, the block is cut and embedded into the ledger. To prevent malicious activity, every signer is allowed to seal a block once per $n/2+1$ blocks, where n is the total number of signers. Thus, there are only $n-(n/2+1)$ signers at a time that can seal a block. Clique consensus does not have immediate finality due to the possibility of creating a fork by proposing two different blocks at a time. Forks occur due to the process called *out-of-order* sealing. It implies that if the current block was not sealed in time, a new block could be proposed by another signer that waited for block period seconds. Out-of-order sealing occurs if block period seconds are configured to be too short for the network configuration, i.e., high latency between nodes and poor network performance. Further, the higher the number of signers, the higher the chance of producing a fork in the blockchain network. Next, since Clique is not BFT, the minimum number of signers for Clique to operate is 1. Finally, in terms of liveness, Clique can tolerate up to one-half of failed signers in a network.

IBFT 2.0 [25] is the Istanbul Byzantine fault tolerant (IBFT) PoA consensus mechanism. It is a variation of Practical BFT [26], which is applicable in blockchain networks. Originally, IBFT 1.0 [27] attempted to bring immediate finality and BFT into the block generation process, which was missing in the Clique consensus. However, Saltini in [28] proved that IBFT 1.0 is not BFT and does not guarantee immediate finality while operating in synchronous networks. This was fixed in IBFT 2.0. Similar to Clique, IBFT 2.0 also has a designated list of signers, called *validators*. IBFT 2.0 achieves immediate finality and prevents the occurrence of forks in the blockchain. However, the minimum number of validators, i.e., quorum, for IBFT 2.0 increased to 4. It achieves quorum and is BFT only if up to $(n-1)/3$ validators are malicious, where n is the total number of validator nodes. IBFT 2.0 achieves consensus in three distinct phases: *pre-prepare*, *prepare*, and *commit*. A new block is disseminated to all validators with a *pre-prepare* message. Then, validators broadcast *prepare* message. When receiving prepared replies from $2/3+1$ of validators and achieving a quorum, the validator broadcasts *commit message*. When *commit* message is received by $2/3+1$ of validators, the new block is written to the ledger. IBFT 2.0 communication complexity is $O(n^2)$. Finally, in terms of liveness, the IBFT 2.0 network can sustain up to one-third of the validators to fail.

QBFT or Quorum BFT [29] is the latest PoA consensus mechanism for HB private networks. It was proposed as a solution to the liveness and safety concerns of IBFT 2.0, i.e., blockchain network DoS when two legitimate validators lock on different blocks. QBFT is similar to IBFT 2.0 in terms of immediate finality, quorum, and liveness. Similar to IBFT 2.0, it has communication complexity of $O(n^2)$ and follows a three-phase commit strategy. However, the difference is that in QBFT, if validators do not achieve consensus before a certain, predefined time expires, the validation round will reset, triggering a new consensus attempt. QBFT achieves immediate finality and prevents the occurrence of forks in the blockchain. Further, it achieves quorum and is BFT only up to $(n-1)/3$ of malicious validator nodes. Finally, in terms of liveness, the QBFT can sustain up to one-third of the validator nodes to fail. QBFT is recommended by HB developers as the enterprise-grade consensus protocol for HB private networks.

3 Marketplace implementation

The energy marketplace is depicted in Fig. 3. It consists of two layers: *physical* and *digital*. The physical layer represents the electricity generation and distribution infrastructure. The digital layer represents the network infrastructure between the energy providers, regulators, and prosumers which enables the electricity trade. This work investigates the digital layer exclusively. Regarding the physical layer, we assume the regulator can correctly map generated electricity to the virtual kWh by combining information from metering devices in the electrical grid with the information stored in the blockchain ledger. Further, the regulator ensures that only certified RES and metering devices are installed in the physical layer of the marketplace.

Each energy provider and regulator are represented within the marketplace as a *blockchain organization* (BO). Each BO must operate at least one validator node. The validator nodes are the main guarantors of valid transaction execution and require the most computational power. Further, each BO has a dedicated *Tessera* node to enable private transaction execution in the network. Finally, each BO has a marketplace interface (MI) that prosumers use to conduct P2P trade settlements.

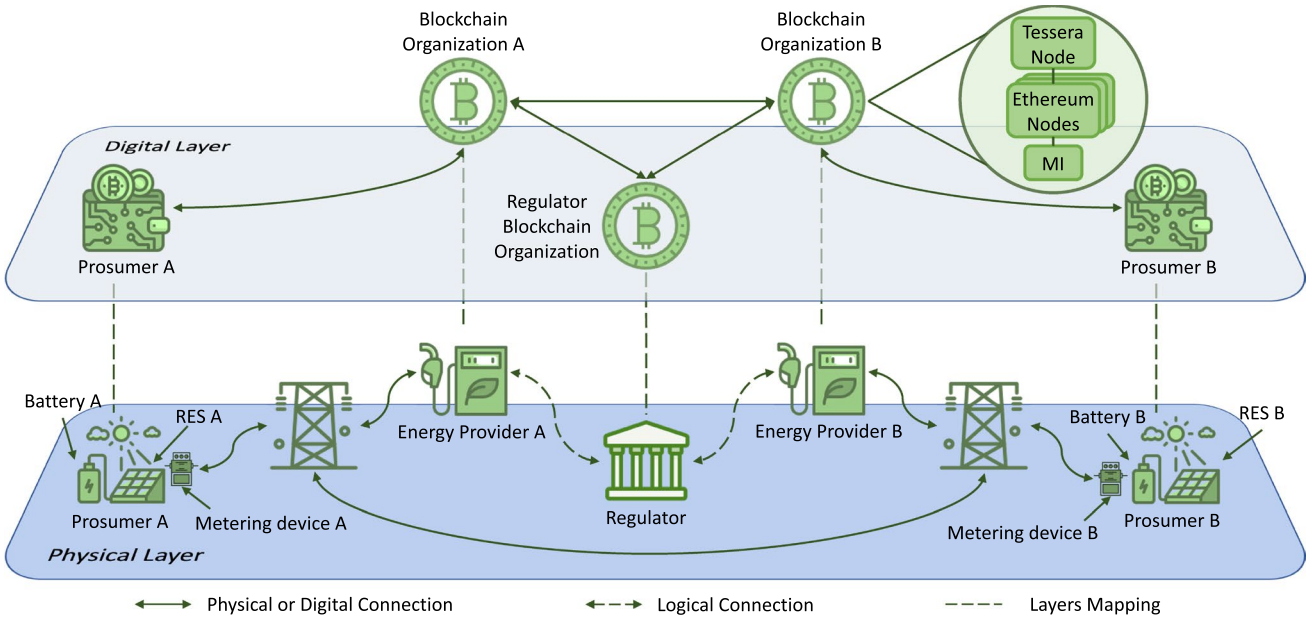

Fig. 3 Energy marketplace (physical layer, i.e., energy grid, is mapped to a digital blockchain-based layer, where the electricity trade operations are executed)

In the marketplace implementation, prosumers are represented as user accounts. Since they do not operate any validators, they need to trust their energy provider's BO to execute transactions on their behalf. Before registration, the prosumer-owned RES must be certified by the regulator. Further, the RES is saved as a data record within PG, which includes the energy provider and regulator. During prosumer registration in the marketplace, the previously created RES record is attached to the prosumer record. In addition, the prosumer receives a personal wallet record where both fiat currency and bought electricity are stored. The energy provider registers the prosumer-generated electricity in the marketplace if the prosumer's RES is marked as certified. While trading, the prosumer utilizes an ordering system where buy or sell orders can be fulfilled according to a predefined marketplace SC.

3.1 Marketplace execution guarantees

Following this setup, HB provides the participants in the blockchain network, i.e., the electricity providers and regulator, with two types of guarantees: (1) the guarantee that the data stored in the ledger cannot be tampered with and (2) the guarantee that it can only be modified following the rules implemented in the SC. These guarantees can be leveraged to fulfill the marketplace requirements. Firstly, by storing the GO in the ledger and encoding the rules governing their life cycle in an SC, i.e., issue and consumption, it is possible to automate their management in a transparent fashion and guarantee that the legislation is followed. Secondly, the same principles can be applied to the management of electricity production, consumption, and trade settlement. By encoding the state of all the marketplace entities in the ledger, i.e., prosumer, RES, and order, the marketplace ensures that there is always a consensus among all participants regarding the status of the marketplace as a whole. Further, by describing all the processes in the marketplace as a set of operations transforming this data and implementing these operations in the SC, it is possible to ensure that all the operations in the marketplace respect the agreed-upon rules.

One limitation of blockchain technology is that it can only provide guarantees after storing the data in the ledger. In other words, it cannot verify the validity of the data inserted in the ledger. In that regard, HB can only provide traceability for the data, recording which actor provided the information. The other marketplace actors must either trust that actor to provide correct information or rely on external processes to verify its validity. Within the marketplace, the regulator is trusted with the insertion of the GO, and the certification of the prosumer-owner RES, and the energy providers are trusted with the report of the energy production. The SC guarantees that the implemented rules are followed for all the other operations. In this case, the challenge is to ensure that the SC implementation matches the legislation. Another limitation appears when designing a system respectful of the privacy of the actors. In this case, the complete state of the system can no longer be publicly stored and shared with all the actors. Instead, it needs to be split, and different parts are then stored in different PGs depending on which actor needs to access the data. Beyond weakening the tamper resistance

Table 2 Prosumer blockchain data record

Field name	Type	Description
ID	*String*	Prosumer's record unique identifier
Electricity	*Double*	Amount of generated electricity *(kWh)*
WalletID	*String*	Prosumer's wallet identifier
RESID	*String*	Prosumer's RES identifier

Table 3 Wallet blockchain data record

Field name	Type	Description
ID	*String*	Wallet's unique identifier
Currency	*Double*	Amount of fiat currency, e.g., USD, EUR
Electricity	*Double*	Amount of prosumer bought electricity *(kWh)*

guarantees, this also introduces additional complexity in the design and implementation of the SC, making it more challenging to ensure that the implementation correctly matches the legislation.

3.2 Marketplace data structure

Each record in HB is saved as $<key, value>$ pairs. *Key* is a unique data identifier and must not repeat within a ledger. *Value* contains data associated with a specific key and all fields that the data record consists of. An underlying data structure is required to manipulate data in trade settlement transactions.

The *prosumer* record is described in Table 2. It is private for the PG which includes the energy provider and regulator. This record contains prosumer unique *ID*. The ID represents the *key* in $<esskey, value>$ pair and contains a user blockchain identity *Address*. The *Electricity* field is updated by the energy provider and regulator based on the data from the prosumer's metering device. Further, it contains an associated wallet and RES IDs. The prosumer record intentionally does not contain any personally identifiable information (PII) to comply with General Data Protection Regulation (GDPR) [30]. All PII needed for legal purposes can be saved in the conventional DB outside of the blockchain.

The *wallet* record is described in Table 3. The *Currency* is the amount of fiat currency the prosumer has. It is used for trade settlement execution. The *Electricity* shows the amount of bought electricity. The wallet record *Electricity* and the prosumer record *Electricity* are separated to ensure that the bought electricity is not resold twice. The wallet record is visible to all energy providers to conduct cross-provider trade settlements.

The GO record is described in Table 4. It is a significant asset that must be presented by the prosumer-seller during the trade settlement execution. The GO records are public for

Table 4 GO blockchain data record

Field name	Type	Description
ID	*String*	GO unique identifier
OwnerID	*String*	GO owner ID
RegulatorID	*String*	Issuer of GO
ElectricityAmount	*Double*	Amount of electricity *(kWh)*
IsConsumed	*Boolean*	Set *True* when electricity is sold

Table 5 Order blockchain data record

Field name	Type	Description
ID	*String*	Order unique identifier
Type	*String*	Order Type *(Sell or Buy)*
Price	*Double*	Price for the entire amount sold
ElectricityAmount	*Double*	Amount of electricity *(kWh)*
GOID	*String*	GO unique identifier
SellerWalletID	*String*	Seller wallet identifier
BuyerWalletID	*String*	Buyer wallet identifier

the entire blockchain network. Further, the GO record contains the respective ids of the prosumer who owns it and the regulator who issued it. Further, *ElectricityAmount* contains the amount of electricity certified by the regulator for further trading. Finally, when the energy is sold, the *isConsumed* field is set *True*.

The *order* record is described in Table 5. *Type* shows what kind of order it is, i.e., sell or buy. Further, the *Price* and *ElectricityAmount* contain the respective amounts of resources required from both parties. The *GOID* links a particular GO to the order. In *buy* order, the *GOID* is left empty to be filled by the seller. The *SellerWalletID* and *BuyerWalletID* fields contain identifiers of prosumer wallets. Depending on the type of the order, when it is created, one of the wallet identifiers is left empty, i.e., *SellerWalletID* is empty for a buy order, and *BuyerWalletID* is empty for a sell order. When the order is fulfilled, it is private for prosumers and energy providers participating in trade settlement.

3.3 Trade settlement smart contract

The implemented SC contains the necessary operations actors require to operate the marketplace and trade electricity. These operations include electricity registration, GO issue and consumption, order creation, and trade settlement. For the purposes of the performance evaluation, this study describes in detail trade settlement SC functions that fulfill the *buy* and *sell* customer electricity orders, c.f., Algorithms 1 and 2. Before the order can be fulfilled, a number of prerequisites have to be met. First, electricity has to be generated and registered within the prosumer's marketplace

```
1: function SellElectricity(Order (Type = Buy), GO, Seller Wallet, Buyer Wallet, SellerID)
2:     if GO.OwnerID == SellerId & GO.IsConsumed == False then
3:         if GO.ElectricityAmount == Order.ElectricityAmount then
4:             if BuyerWallet.Currency ≥ Order.Price then
5:                 Order.SellerWalletID ← SellerWallet.ID
6:                 Order.GOID ← GO.ID
7:                 SellerWallet.Currency ← SellerWallet.Currency + Order.Price
8:                 BuyerWallet.Currency ← BuyerWallet.Currency − Order.Price
9:                 BuyerWallet.Electricity ← BuyerWallet.Electricity + Order.ElectricityAmount
10:                Commit
11:            else
12:                return Insufficient Buyer Currency.
13:        else
14:            return Insufficient Electricity Amount.
15:    else
16:        return Invalid GO Attached to the Order.
   Execute FinalizeOrder(Order, GO)                                   ▷ c.f., Algorithm 3.
```

Algorithm 1 Fulfill buy electricity order

```
1: function BuyElectricity(Order (Type = Sell), GO, BuyerWallet, SellerWallet)
2:     if BuyerWallet.Currency ≥ Order.Price then
3:         if GO.IsConsumed == False then
4:             Order.BuyerWalletID ← BuyerWallet.ID
5:             SellerWallet.Currency ← SellerWallet.Currency + Order.Price
6:             BuyerWallet.Currency ← BuyerWallet.Currency − Order.Price
7:             BuyerWallet.Electricity ← BuyerWallet.Electricity + Order.ElectricityAmount
8:             Commit
9:         else
10:            return Invalid GO Attached to the Order.
11:    else
12:        return Insufficient Buyer Currency.
   Execute FinalizeOrder(Order, GO)                                   ▷ c.f., Algorithm 3.
```

Algorithm 2 Fulfill sell electricity order

```
1: function FinalizeOrder(Order, GO)
2:     GO.IsConsumed ← True
3:     Delete(Order)
4:     Commit
```

Algorithm 3 Finalize order

account. Further, a GO has to be issued for the amount of electricity that is being sold. Finally, the order itself has to be created. For both order types, the trade settlement operation execution has two stages. This is required due to HB SC's inability to modify private, i.e., wallets and order, and public, i.e., GO, data in a single transaction.

In the first stage of Algorithm 1, the *SellElectricity* function takes a *buy* order posted by a prosumer-buyer. Further, the algorithm performs a number of security checks. Since the prosumer-seller provides the GO at the moment of trade settlement execution, it is verified to have the correct ownership. Further, the GO's *IsConsumed* field is checked to be *False*. Finally, the GO is checked to be issued for the amount of electricity listed in the buy order. Next, the buyer's wallet is verified to have an appropriate currency to buy the electricity. Finally, the resources are exchanged between the buyer and seller, i.e., electricity and currency. This transaction is private for PG, which includes trading prosumers' energy providers and the regulator.

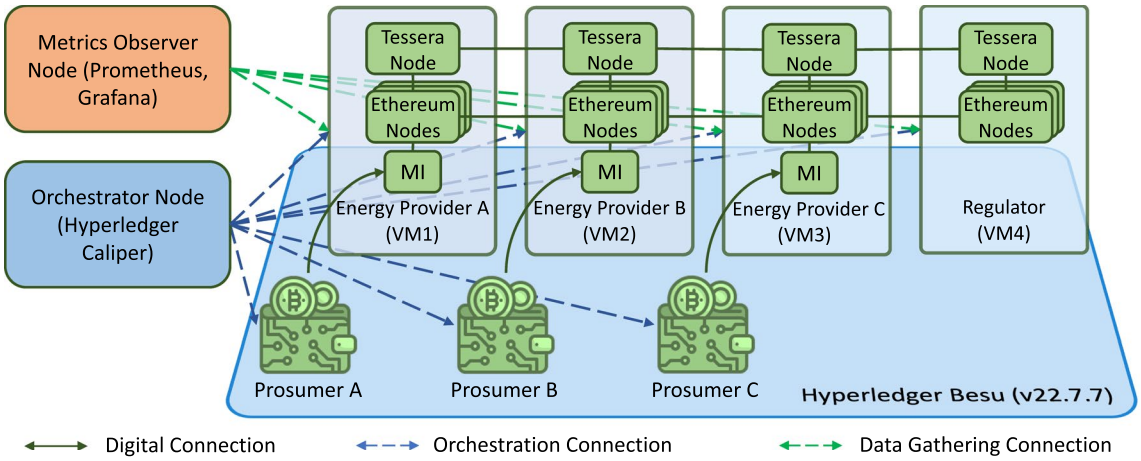

Fig. 4 Implemented energy marketplace

In the first stage of Algorithm 2, the *BuyElectricity* function takes a *sell* order posted by a prosumer-seller. Further, the algorithm verifies if the GO is consumed and if the buyer has enough currency in the wallet. Finally, the resources are exchanged between the buyer and seller. The correct ownership of the GO and conformity of GO's and order's electricity amounts is not checked in *BuyElectricity* function. These checks are performed during sell order creation, i.e., GO has to be provided during the sell order creation.

In the second stage of both Algorithms 1 and 2, the *FinalizeOrder* function is executed by the buyer's energy provider, i.e., the actor interested in preventing electricity double-spending. First, this function takes the GO, sets its *IsConsumed* value to *True*, and saves it in the public ledger. Further, it marks the fulfilled order as deleted. Thus, the order is not visible in the order chart but can be found in the ledger history.

4 Performance evaluation

The throughput of public transactions, i.e., visible to the entire private network, has already been investigated by the authors of [31]. The main aim of this study is to measure the performance of private transaction execution with the SC tailored to the energy marketplace needs. The performance evaluation was conducted on the test infrastructure described in Fig. 4. The infrastructure consists of 4 virtual machines (VMs), where each VM size is 16 vCPUs, 64 GB RAM, and 256 GB high throughput (150 MB/s) disk space. Energy providers A, B, and C run VM1, VM2, and VM3, respectively, while the regulator runs VM4. All VMs are connected with a 10Gbit/s network interface. In our experimental implementation, we use HB version 22.7.7 and Tessera 22.1.7 without modifying the core code. All nodes within the infrastructure are deployed as docker containers. To collect reliable and correct performance evaluation data, *Prometheus*,[5] *Grafana*,[6] and *Hyperledger Caliper*[7] (HC) tools are utilized. The *Prometheus* is used as the main blockchain operation data collector. The *Grafana* is used as a data visualization tool. The *HC* performance evaluation tool is used as a transaction load generator. In Fig. 4, the *HC* is part of the Orchestrator Node and is located on a separate VM. The *HC* executes transactions bypassing the MI. Thus, the MI is not a part of the performance evaluation.

Several performance metrics are considered in this study. First, the *throughput* is the number of successful transactions (TPS) or reads (RPS) executed per second in the blockchain network. The *latency* is the time it takes to finalize transaction execution and write it to the ledger or return a reply with the query result. The *scalability* is the behavior of the network with an increasing number of nodes. Scalability is also dependent on the size of PG.

This study manipulated several configuration parameters within HB to investigate the maximum throughput. These parameters were selected based on the performance tests conducted by the HB developers and research studies [31]. The *block period seconds* (BPS) parameter defines the time validators accept transactions to add to the new block. When the BPS time is up, the block is cut and embedded into the ledger. Further, horizontal scalability is investigated by changing the number of validator nodes and PG size. To investigate write transaction throughput, 5-min tests were executed with a constant send rate. To investigate read throughput, the 4-kb asset was read from the local

[5] https://prometheus.io/docs/introduction/overview/

[6] https://grafana.com/

[7] https://hyperledger.github.io/caliper/

Table 6 Performance evaluation parameters

Parameter	Value
Transaction send rate (write)	10, 20 → 300 with step of 20 *(fixed rate in duration of 5 min)
Block period seconds (BPS)	1 → 6 with step of 1
Transaction send rate (read)	100, 300 → 3000 with step of 300 *(fixed rate in duration of 5 min)
Validator nodes	4 → 24 with step of 4
Privacy group size	2, 3, 4
Consensus mechanism	Clique, IBFT 2.0, QBFT

HB database, i.e., state database, with varying query send rates. Table 6 summarizes the entire performance evaluation parameters configuration.

4.1 Write–trade settlement execution

In this study, an Algorithm 1 was executed as an SC function to test maximum write TPS. An Algorithm 1 was chosen as a load generator due to having the highest computational complexity out of all defined SC functions. To write a transaction to the ledger, a respective consensus mechanism, i.e., Clique, IBFT 2.0, or QBFT, must be executed. First, we test the baseline HB configuration, which included the minimum necessary setup to operate, i.e., four validators, BPS = 1 s. The PG size is 3, i.e., energy providers A and B, and the regulator. The throughput measurement results are shown in Fig. 5. All consensus mechanisms show a similar performance of approximately 200 TPS. However, QBFT demonstrated the best latency. Clique and IBFT 2.0 demonstrate higher latency both for peak throughput and further increase of send rate exceeding the maximum TPS. The baseline test demonstrates the maximum sustainable network load with private transactions of around 200 TPS. Thus, further tests are conducted with a fixed send rate of 200 TPS.

Next, the maximum TPS with a varying BPS was investigated, c.f., Fig. 6. The results demonstrate that the BPS affects the maximum throughput of the HB network, i.e., the BPS increase results in a steady throughput decrease. Further, the latency rises significantly, e.g., up to approximately 6-s latency for BPS = 6 s. Here all investigated consensus mechanisms show similar performance under varying BPS, where QBFT is the best performer. The results demonstrate that BPS and eventual latency increase significantly affecting the quality of service (QoS) [32] provided by the marketplace. QoS is aimed to maximize the user experience in terms of response time and transaction success rate by addressing throughput and scalability issues. In this case, the BPS has to be considered an important metric for QoS provisioning [33].

The horizontal scalability was investigated with varying validators number and a PG size. The results of the validator scalability investigation are shown in Fig. 7. Here, the number of validator nodes was changed from 4 to 24 with a step of 4. Results demonstrate that the number of validator nodes significantly affects the maximum network throughput. It represents a significant performance bottleneck resulting in approximately 42% throughput reduction with 24 validators. Further, the latency increases significantly, reaching approximately 4.5 s for IBFT 2.0. Here, all investigated consensus mechanisms demonstrate similar performance, with QBFT having the highest TPS and the lowest latency. In addition, QBFT demonstrates the best scalability by maintaining 190–200 TPS up to 12 validators.

The results of PG size scalability are shown in Fig. 8. The investigated PG sizes are under four nodes because each BO can operate only one Tessera node, i.e., this is an infrastructure limitation. The PG size increase does not result in a significant throughput decrease. However, the latency increases approximately by a half second for all investigated consensus mechanisms with a PG size equal to 4. Here, the performance of consensus mechanisms is similar, with QBFT showing the best results.

The performance evaluation results demonstrate that the maximum possible throughput depends significantly on BPS and network size, i.e., the best throughput is achieved with BPS = 1 s and 4 validators configuration. Further, the QBFT has the best throughput, latency, and scalability characteristics out of all investigated consensus mechanisms. Finally, the performance evaluation shows that the HB-based marketplace demonstrates an approximately two times lower throughput and higher latency than the HF-based marketplace investigated in [34]. However, HF uses the RAFT consensus mechanism, which is only CFT, i.e., does not protect from malicious nodes. In contrast, HB's IBFT 2.0 and QBFT consensus mechanisms are BFT, i.e., protect the blockchain network for up to one-third of malicious nodes at the cost of increased computational complexity.

4.2 Ledger data read

In order to write any data to the blockchain ledger, a consensus mechanism has to be executed. The query, i.e., read, request does not execute a consensus mechanism to get the requested data. Thus, the block or network configuration, i.e., validator number, does not affect the read throughput. Here, it is the amount of data, i.e., asset size, that is read from the individual blockchain node that affects reads per second (RDS). The read throughput is shown in Fig. 9. To investigate read throughput, the query was constructed to read 4-kb of data from the RockDB world state database.

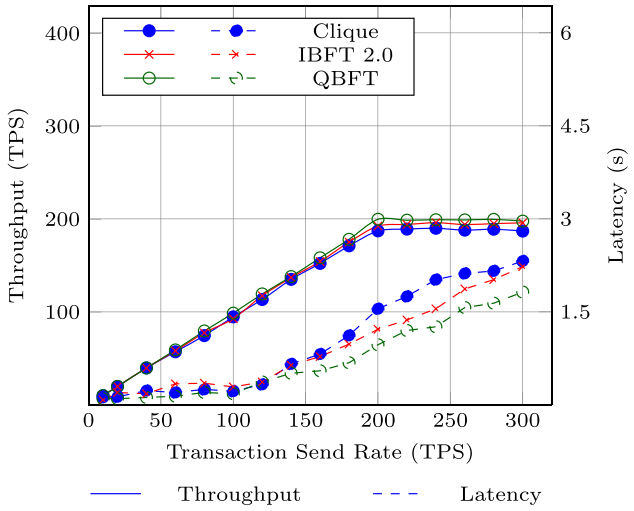

Fig. 5 Transaction throughput and latency with varying send rate (BPS = 1 s, validators = 4, PG size = 3)

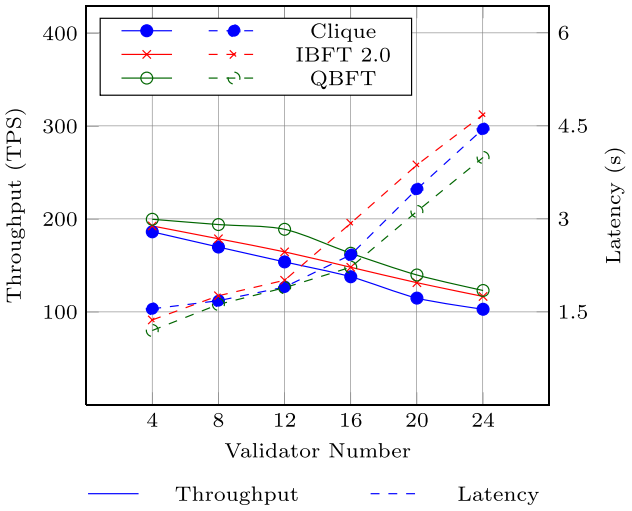

Fig. 7 Throughput and latency with varying validator nodes number (BPS = 1 s, 200 TPS send rate)

The results demonstrate the maximum throughput of approximately 1440 RPS for all investigated consensus mechanisms. The latency remains under 500 ms until the moment we reach peak read throughput. From there on, the latency starts to increase rapidly if the read queries send rate is over peak throughput.

The read throughput evaluation results show that the HB-based energy marketplace demonstrates a similar throughput to the HF-based marketplace investigated in [34]. This is an indicator that world state databases used in HB and HF; i.e., RockDB and LevelDB, respectively, demonstrate similar performance when it comes to reading assets of similar size.

5 Results and observations

During the design, implementation, and performance evaluation of the proposed marketplace, a number of observations and conclusions were made, which bring an enhanced understanding of the advantages and limitations of HB. Such observations and conclusions are discussed next.

5.1 Limitations of private transaction execution

System transaction throughput is an important performance characteristic that must be considered in the system design phase. If throughput requirements are not met, the production

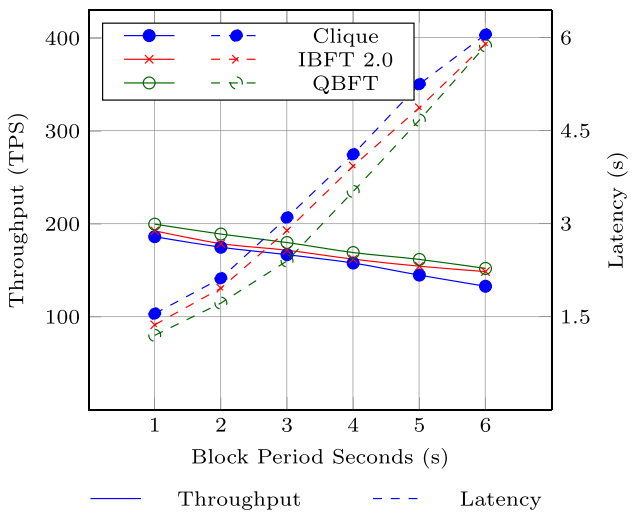

Fig. 6 Throughput and latency with varying block period seconds (200 TPS send rate)

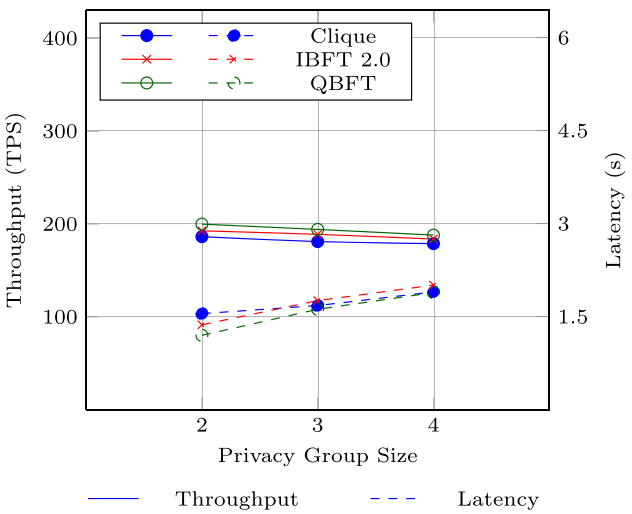

Fig. 8 Throughput and latency with varying PG size (BPS = 1 s, validators = 4, 200 TPS send rate)

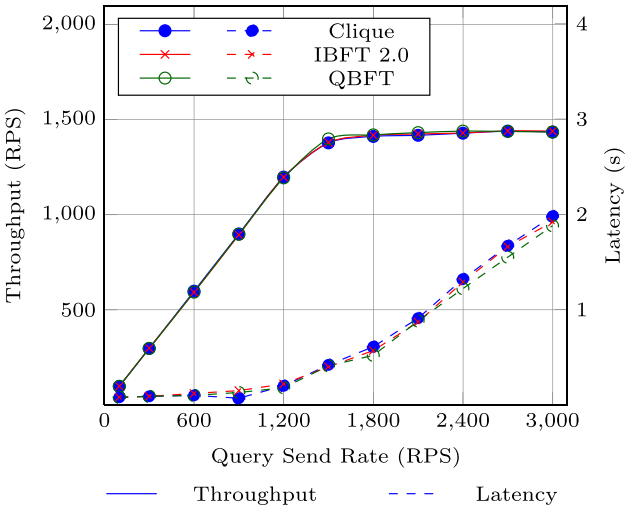

Fig. 9 Read throughput and latency (4-kb asset)

system QoS and scalability requirements will be impossible to meet. Different blockchain architectures, i.e., public and private, demonstrate a considerable performance difference in terms of throughput and latency. Higher system decentralization and security come at a cost of additional computational complexity. Performance evaluation of the HB-based energy marketplace demonstrates a throughput of approximately 200 TPS which is a considerable improvement in comparison with public blockchains such as Bitcoin or Ethereum [21]. However, the HB still requires an improvement in throughput and scalability to get to the level of performance demonstrated by HF.

The measured HB throughput of 200 TPS can be used to estimate the maximum number of prosumers the energy marketplace can support [34]. However, this estimation is a first-order approximation and may not account for other factors that can affect the maximum number of prosumers in the real-world system. Several metrics were identified to make this estimation, including the maximum throughput of the system T_{max}, the number of daily electricity generation registrations m, the number of orders created by a prosumer per day o, the number of trade transactions executed per day t, and the number of GOs issued for a prosumer per day g. Metric t was identified as a full number of blockchain transactions needed to finalize energy trade. As trade settlement has two stages, its every execution requires two blockchain transactions. Dividing the maximum daily amount of transactions by the sum of m, o, t, and g yields an approximate number of prosumers that can operate within the marketplace, c.f., Eq. (1).

$$Prosumers_{max} = \frac{T_{max} * 24 * 60 * 60}{m + o + t + g} \quad (1)$$

The throughput of 200 TPS implies that the energy marketplace can execute 17,280,000 transactions in 24 h. The configuration of the m, o, t, and g determines the maximum number of prosumers the marketplace can support. The m needs to be set first because it affects the rest of the parameters. If the m parameter is set to 24, which means hourly registrations, the prosumer can trade their generated electricity 23 times daily (minus 1 h for initial generation). This worst-case scenario assumes the prosumer trades every time the metering device updates. To execute a trade, the prosumer must place an order and have a GO issued. In this scenario, the total number of daily prosumer transactions equals 116 (24 + 23 + 23*2 + 23). Therefore, the maximum number of prosumers supported by the marketplace is 17,280,000 / 116 ≈ 150,000, corresponding to a small-to-medium-size energy community.

5.2 Limited auditability of private transactions

The auditability and integrity of all data in the decentralized network are characteristics that affect the guarantees that the system can provide for its users. However, private transactions imply that only a portion of blockchain network participants see the contents and participate in consensus execution for a particular transaction. As the Ethereum blockchain was not designed to work with private transactions, the *Tessera* private transaction manager was adopted in HB. It is built as a separate entity and complements the implementation of the Ethereum Enterprise Client. However, it comes at the cost of private transactions' auditability. The Tessera nodes distribute private transactions to the members of PG. However, the rest of the nodes outside of PG receive the record confirming that the private transaction was executed. Such an approach results in a limitation where the blockchain network participants outside of PG cannot verify the validity of the private transaction data. This is a result of the inability of non-PG members of the HB network to verify the correctness of private SC deployment and transaction execution.

5.3 Public and private data modification

The integrity of data within a marketplace relies heavily on the correctly defined SC. In the case of private transaction execution, the data within the blockchain is split into public, i.e., seen by all network participants, and private, i.e., available only to the members of PG. However, in a function such as an energy trade settlement, we need to modify both public and private data within one operation. The HF allows such modifications within one transaction without the exposure of private data to non-PG members. In contrast, the HB does not have such a capability, and modification of public and private data has to be split into two different

transactions. Such a limitation opens up additional security concerns where the delay between private and public data modification transactions may be used to disrupt trade process execution or attempt double spending of GOs.

5.4 Private blockchain lesser energy consumption

The excessive amount of computations needed for the transaction execution within a blockchain network raised concern amount environmentalists regarding subsequent carbon emissions [35]. Blockchain implementations such as Bitcoin with PoW consensus algorithm require the execution of computationally heavy tasks, which in the long-term perspective may lead to the carbonization of Earth's atmosphere and cause harmful effects on humankind. Bitcoin is representative of public blockchain architecture, which typically involves a vast number of computing machines involved in blockchain operation and transaction generation. Such public blockchains enable decentralized environments that provide digital sovereignty to their users [36]. The HB, as well as HF, is a representative of private blockchain architecture, which is typically deployed for a targeted business use case that involves selected actors. Consequently, private blockchains assume a certain degree of centralization within a blockchain system, which requires collaborating entities to have certain legal agreements outside of blockchain guarantees, i.e., in the case of the energy marketplace, it is the reliance on TTP such as the regulator. However, private blockchain deployments with PoA consensus mechanisms consume a small fraction of computations when compared to Bitcoin's PoW. Thus, private blockchains are more sustainable in a long-term approach [37].

6 Related work

Hyperledger Foundation has created several projects which employ different blockchain architectures, i.e., public and private, to address industrial and business use cases [38]. Thus, private blockchains like HF and HB became the main energy marketplace implementation and investigation tools. Recently, there were a number of proposals on blockchain-based energy marketplaces in terms of system architecture, electricity trading framework, and performance evaluation. Such proposals are discussed next.

In [39], the authors propose an HB-based P2P marketplace for energy trading and payment settlement. The marketplace utilizes HB as a blockchain platform and IBFT 2.0 as a consensus mechanism. Further, the authors compare IBFT 2.0 with Clique, PoW, and HF's RAFT. According to the authors, their marketplace demonstrates better throughput and latency than PoW and Ethereum Clique. Further, the authors claim that the proposed unified energy trading model provides lower latency compared to similar systems based on PoW, Clique, and HF's RAFT. The authors of [40] propose an HF-based P2P energy marketplace for tokenized energy assets. Such assets are traded within the marketplace, where each actor can benefit monetarily depending on its role. Further, the authors define actors and requirements for the P2P energy marketplace. However, their marketplace does not include a regulator role, GO usage, and data privacy requirements intrinsic to energy market systems. The authors claim their implementation achieved a throughput of 448.3 TPS with transactions that modify public data. However, the authors do not consider private transaction execution and PG throughput impact. In [41], the authors propose an automated blockchain-based P2P energy marketplace based on a multi-agent system paradigm. Permissioned blockchain allows for reduced transaction costs, enables marketplace micro-transactions, and eliminates a single point of failure. According to the authors, blockchain technology enables prosumer self-sovereignty while allowing the marketplace to comply with current data regulations. The authors of [42] propose an HB-based framework for P2P energy trading. The proposed marketplace uses a flexible permission ascription scheme that utilizes HB permissioning and IBFT 2.0 consensus mechanism. According to the authors, the proposed framework provides an efficient scheme for P2P energy trading compared to other solutions. The authors claim that IBFT 2.0 has five times lower latency than Ethereum PoW and two times lower than HF RAFT and KAFKA. Further, performance evaluation demonstrated that IBFT 2.0 has 1.5 times higher throughput than HF's RAFT and Kafka and three times higher than Ethereum PoW. In [43], the authors propose an HF-based platform for the transactive energy marketplace. A proposed platform has a layered architecture consisting of physical, communications, and blockchain layers. Further, the authors use the energy generation data from a real-world energy provider and build its digital twin as a physical layer for their platform. The authors claim that the developed prototype allows trading electricity via SCs developed within the HF network. The authors of [44] propose a blockchain-based marketplace platform that enables energy trading between institutions and electric vehicle (EV) owners. Within the case study, institutions own RES and sell generated electricity to EV owners via a P2P trade contract. The authors claim that such a marketplace platform enables synergy between institutions and EV owners, providing clean and affordable energy. For further reading on the developments in blockchain-based energy marketplaces, the reader is referred to [45].

In [31], the authors conduct an in-depth performance evaluation of the HB platform and its three main consensus mechanisms for private blockchain, i.e., Clique, IBFT 2.0, and QBFT. According to the authors, the performance of HB has a number of bottlenecks, such as transaction execution

and blockchain state updates, which are influenced by node computation power and transaction complexity. Authors claim that QBFT consensus has the best performance and scalability results, achieving a write throughput of approximately 450 TPS and scalability of up to 14 validator nodes. The authors of [46] compare the main proof-based consensus mechanisms, focusing on security and performance. The authors highlight the centralization tendency and the vulnerabilities of main proof-based consensus mechanisms, i.e., PoW, PoS, PoA, and Delegated PoS (DPoS). According to the authors, DPoS consensus has the best balance between throughput, latency, and scalability. However, such a balance comes at the cost of increased centralization and reduced protection against malicious activity.

The related work demonstrate that the application of blockchain technology in the context of an energy marketplace has been defined at a level of abstract entities and operations. However, all works mentioned above lack requirements definition and alignment with the existing regulation on P2P energy trading. Further, the related works lack implementation details and discussion on the technical limitations of blockchain technology incorporation. This work discusses the energy marketplace from regulatory and technical perspectives to provide insights into challenges encountered during the implementation of private transactions execution and system operation.

7 Summary and outlook

This work proposes a decentralized blockchain-based P2P energy marketplace that addresses actors' privacy and the performance of consensus mechanisms. The main aim of the marketplace is to automate the P2P trade settlement process while preserving actors' privacy. The novelty of the proposed marketplace is its alignment with the current energy trade regulations defined in D2018/2001 of the European Parliament. More specifically, our marketplace incorporates the *regulator* actor. The regulator represents a governmental authority that controls renewable energy trading via GO issue and price regulation. In addition, the regulator certifies the RES used to generate traded electricity. Hence, with current regulations, the marketplace is partially centralized around the regulator actor but still improves the automation of energy trading.

Performance evaluation results of an HB-based marketplace private transaction execution with three main consensus mechanisms, i.e., Clique, IBFT 2.0, and QBFT, demonstrate a throughput of approximately 200 TPS with baseline configuration. The QBFT consensus mechanism shows the best throughput and latency. Further, QBFT demonstrates the best scalability by maintaining 190–200 TPS throughput for up to 12 validators. However, HB's QBFT consensus mechanism demonstrates lower throughput than another popular private permissioned blockchain platform HF. This is a side effect of BFT and, thus, increased computations of QBFT. In contrast, HF executes the RAFT consensus mechanism, which is CFT, i.e., more centralized and vulnerable to collusion between malicious nodes. However, the inherent centralization around the regulator mitigates this issue, making HF better suited for such a use case.

Future work will focus on investigating possible improvements for consensus mechanisms for blockchain-based marketplaces to improve the efficiency of their operation. In addition, investigating a trade-off between the performance and security of private blockchains is of interest.

Acknowledgements The project partners in Symphony are Ericsson AB (Stockholm, Sweden) and Affärsverken Energi AB (Karlskrona, Sweden).

Funding Open access funding provided by Blekinge Institute of Technology. The work was partly sponsored by the Swedish Knowledge Foundation through the project *Symphony—Supply-and-Demand-based Service Exposure using Robust Distributed Concepts*.

Data Availability The experiments conducted in this work did not use any predefined data packages.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Yang Y, Zhang S, Xiao Y (2015) Optimal design of distributed energy resource systems coupled with energy distribution networks. Energy 85:433–448. https://doi.org/10.1016/j.energy.2015.03.101
2. Jasim B, Taheri P (2018) An origami-based portable solar panel system. In: 2018 IEEE 9th Annual Information Technology. Electronics and Mobile Communication Conference (IEMCON), IEEE, pp 199–203. https://doi.org/10.1109/IEMCON.2018.8614997
3. Pop C, Cioara T, Antal M, Anghel I, Salomie I, Bertoncini M (2018) Blockchain based decentralized management of demand response programs in smart energy grids. Sensors 18(2):1–21. https://doi.org/10.3390/s18010162

4. EU Parliament (2018) Directives Directive (EU) 2018/2001 of the European Parliament, pp 82–209. http://data.europa.eu/eli/dir/2018/2001/2022-06-07. Accessed 18 June 2023
5. Hamburger A (2019) Is guarantee of origin really an effective energy policy tool in Europe? A critical approach. Soc Econ 41(4):487–507. https://doi.org/10.1556/204.2019.41.4.6
6. Hertz-Shargel B, Livingston D, Atlantic Council of the United States (2019) Assessing blockchain's future in transactive energy. https://www.atlanticcouncil.org/in-depth-research-reports/report/assessing-blockchains-future-in-transactive-energy/. Accessed 18 June 2023
7. Kollmann T, Hensellek S, de Cruppe K, Sirges A (2020) Toward a renaissance of cooperatives fostered by blockchain on electronic marketplaces: a theory-driven case study approach. Electron Mark 30(2):273–284. https://doi.org/10.1007/s12525-019-00369-4
8. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. Accessed 18 June 2023
9. Singh J, Michels JD (2018) Blockchain as a Service (BaaS): providers and trust. In: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS & PW). IEEE, pp 67–74. https://doi.org/10.1109/EuroSPW.2018.00015
10. Hyperledger Foundation (2022) Hyperldger Besu Ethereum client. https://besu.hyperledger.org/en/stable/. Accessed: 18 June 2023
11. Tkachuk R-V, Ilie D, Robert R, Kebande V, Tutschku K (2023a) On the performance of consensus mechanisms in privacy-enabled decentralized peer-to-peer renewable energy marketplace. In: 26th Conference on Innovation in Clouds. Internet and Networks and Workshops (ICIN), IEEE, pp 179–186. https://doi.org/10.1109/ICIN56760.2023.10073510
12. Akay H, Kim SG (2021) Reading functional requirements using machine learning-based language processing. CIRP Annals 70(1):139–142. https://doi.org/10.1016/j.cirp.2021.04.021
13. Wilson KB, Karg A, Ghaderi H (2022) Prospecting non-fungible tokens in the digital economy: stakeholders and ecosystem, risk and opportunity. Bus Horiz 65(5):657–670. https://doi.org/10.1016/j.bushor.2021.10.007
14. Wang Q, Li R, Wang Q, Chen S (2021) Non-Fungible Token (NFT): Overview, evaluation, opportunities and challenges. https://arxiv.org/abs/2105.07447
15. Khatter K, DevanjaliRelan (2022) Non-functional requirements for blockchain enabled medical supply chain. Inte J Syst Assur Eng Manag 13(3):1219–1231. https://doi.org/10.1007/s13198-021-01418-y
16. Qazi A, Hussain F, Rahim NABD, Hardaker G, Alghazzawi D, Shaban K, Haruna K (2019) towards sustainable energy: a systematic review of renewable energy sources, technologies, and public opinions. IEEE Access 7:63837–63851. https://doi.org/10.1109/ACCESS.2019.2906402
17. Binjubeir M, Ahmed AA, Ismail MAB, Sadiq AS, Khan MK (2020) Comprehensive survey on big data privacy protection. IEEE Access 8:20067–20079. https://doi.org/10.1109/ACCESS.2019.2962368
18. Xie J, Tang H, Huang T, Yu FR, Xie R, Liu J, Liu Y (2019) A survey of blockchain technology applied to smart cities: research issues and challenges. IEEE Commun Surv Tutor 21(3):2794–2830. https://doi.org/10.1109/COMST.2019.2899617
19. Tkachuk R-V, Ilie D, Tutschku K, Robert R (2021) A survey on blockchain-based telecommunication services marketplaces. IEEE Trans Netw Serv Manag 19(1):228–255. https://doi.org/10.1109/TNSM.2021.3123680
20. Liu M, Wu K, Xu JJ (2019) How will blockchain technology impact auditing and accounting: permissionless versus permissioned blockchain. Curr Issues Audit 13(2). https://doi.org/10.2308/ciia-52540
21. Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger. In: Ethereum project yellow paper, pp 1–32. https://gavwood.com/paper.pdf.
22. Kaleem M, Anastasia M, Aron L (2020) Vyper: a security comparison with solidity based on common vulnerabilities. In: 2020 2nd conference on Blockchain Research and Applications for Innovative Networks and Services (BRAINS), pp 107–111. https://doi.org/10.1109/BRAINS49436.2020.9223278
23. Vukolić M (2016) The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Camenisch J, Kesdoğan D (eds) Open problems in network security. iNetSec 2015, Lecture Notes in Computer Science(), vol 9591. Springer, Cham, pp 112–125. https://doi.org/10.1007/978-3-319-39028-4_9
24. Szilágyi P (2017) EIP-225: Clique proof-of-authority consensus protocol. https://eips.ethereum.org/EIPS/eip-225. Accessed 18 June 2023
25. Saltini R, Hyland-Wood D (2019b) IBFT 2.0: A safe and live variation of the IBFT blockchain consensus protocol for eventually synchronous networks. https://arxiv.org/abs/1909.10194
26. Castro M, Liskov B (1999) Practical Byzantine Fault Tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI '99). USENIX Association, New Orleans, pp 173–186
27. Lin Y-T (2017) Istanbul byzantine fault tolerance. https://github.com/ethereum/EIPs/issues/650. Accessed 18 June 2023
28. Saltini R, Hyland-Wood D (2019) Correctness analysis of IBFT. https://arxiv.org/abs/1901.07160
29. Moniz H (2020) The Istanbul BFT consensus algorithm. https://arxiv.org/abs/2002.03613
30. EU Parliament (2016) Regulation (EU) 2016/679 of the European Parliament (General Data Protection Regulation). In: General data protection regulation, pp 1–99. https://gdpr-info.eu/
31. Fan C, Lin C, Khazaei H, Musilek P (2022) Performance analysis of Hyperledger Besu in private blockchain. In: 2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), pp 64–73. https://doi.org/10.1109/DAPPS55202.2022.00016
32. Chen W, Paik I (2015) Toward better quality of service composition based on a global social service network. IEEE Trans Parallel Distrib Syst 26(5):1466–1476. https://doi.org/10.1109/TPDS.2014.2320748
33. Vaghani A, Sood K, Yu S (2022) Security and QoS issues in blockchain enabled next-generation smart logistic networks: A tutorial. Blockchain: Res Appl 3(3):1–14. https://doi.org/10.1016/j.bcra.2022.100082
34. Tkachuk R-V, Dragos I, Remi R, Kebande V, Tutschku K (2023b) Towards efficient privacy and trust in decentralized blockchain-based peer-to-peer renewable energy marketplace. Sustain Energy Grids Netw 1–27. https://doi.org/10.1016/j.segan.2023.101146
35. Sedlmeir J, Ulrich Buhl H, Fridgen G, Keller R (2021) Recent developments in blockchain technology and their impact on energy consumption. Informatik Spektrum 43(6):391–404. https://doi.org/10.1007/s00287-020-01321-z
36. Morrow MJ, Zarrebini M (2019) Blockchain and the tokenization of the individual: societal implications. Future Internet 11(10):1–12. https://doi.org/10.3390/fi11100220
37. Bada AO, Damianou A, Angelopoulos CM, Katos V (2021) Towards a green blockchain: a review of consensus mechanisms and their energy consumption. In: 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, pp 503–511. https://doi.org/10.1109/DCOSS52077.2021.00083
38. Li D, Wong EW, Guo J (2020) A survey on blockchain for enterprise using hyperledger fabric and composer. In: 2019 6th International Conference on Dependable Systems and Their Applications (DSA), pp 71–80. https://doi.org/10.1109/DSA.2019.00017
39. Abdella J, Tari Z, Anwar A, Mahmood A, Han F (2021) An architecture and performance evaluation of blockchain-based

peer-to-peer energy trading. IEEE Trans Smart Grid 12(4):3364–3378. https://doi.org/10.1109/TSG.2021.3056147
40. Karandikar N, Chakravorty A, Chunming Rong C (2021) Blockchain based transaction system with fungible and non-fungible tokens for a community-based energy infrastructure. Sensors 21(11):1–32. https://doi.org/10.3390/s21113822
41. Mezquita Y, Gil-González AB, Martín del Rey A, Prieto J, Corchado JM (2022) Towards a blockchain-based peer-to-peer energy marketplace. Energies 15(9):3046. https://doi.org/10.3390/en15093046
42. Pradhan NR, Singh AP, Kumar N, Hassan MM, Roy DS (2022) A flexible permission ascription (FPA)-Based blockchain framework for peer-to-peer energy trading with performance evaluation. IEEE Trans Ind l Inform 18(4):2465–2475. https://doi.org/10.1109/TII.2021.3096832
43. Boumaiza A, Wanik MZC, Sanfilippo A (2022) Modeling a blockchain-enabled transactive energy system for community microgrids. In: 2022 IEEE 16th International Conference on Compatibility, Power Electronics, and Power Engineering (CPE-POWERENG), pp 1–6. https://doi.org/10.1109/CPE-POWERENG54966.2022.9880874
44. Cavalcante I, Jamilson J, Manzolli JA, Almeida L, Pungo M, Guzman CP, Morais H (2023) Electric vehicles charging using photovoltaic energy surplus: a framework based on blockchain. Energies 16(6):2694. https://doi.org/10.3390/en16062694
45. Choobineh M, Arabnya A, Sohrabi B, Khodaei A, Paaso A (2023) Blockchain technology in energy systems: A state-of-the-art review. IET Blockchain 3(1):35–39. https://doi.org/10.1049/blc2.12020
46. Rebello GAF et al (2022) A security and performance analysis of proof-based consensus protocols. Ann Telecommun 77(7-8):517–537. https://doi.org/10.1007/s12243-021-00896-2

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.