The Role of Blockchains in Multi-Stakeholder Transactive Energy Systems

Scott Eisele
Vanderbilt University

Aron Laszka
University of Houston

Douglas C. Schmidt
Vanderbilt University

Abhishek Dubey
Vanderbilt University

***Abstract*— Power grids are undergoing major changes due to rapid growth in renewable energy and improvements in battery technology. Prompted by the increasing complexity of power systems, decentralized solutions are emerging that arrange local communities into transactive microgrids. This paper addresses the problem of implementing transactive energy mechanisms in a distributed setting, providing both privacy and safety. Specifically, we design and implement an automated auction and matching system that ensures safety (e.g., satisfaction of line capacity constraints), preserves privacy, and promotes local trade and market efficiency for transactive energy systems. This design problem is challenging because safety, market efficiency, and privacy are competing objectives. We implement our solution as a decentralized trading platform built on blockchain technology and smart contracts. To demonstrate the viability of our platform, we analyze the results of experiments with dozens of embedded devices and energy production and consumption profiles using an actual dataset from the transactive energy domain.**

***Index Terms*—Transactive energy systems, blockchain, privacy, security, safety, smart contract.**

I. Introduction

Current energy markets are competitive wholesale markets that are run by *Regional Transmission Organizations* (RTO). An RTO typically runs two energy markets: a *day-ahead market*, which occurs 24 hours prior to energy dispatch, and a *real-time market*, which is run once each hour based on predicted energy demand. Any adjustments necessary to meet changing energy demands are handled by ancillary services, such as *Reserves*, which can be integrated with the grid in under an hour, and *Regulation*, which maintains system frequency by adjusting output. In this market, power plants and utility operators place offers that are matched to determine pricing.

Due to the advent of individually-owned *distributed energy resources* (DER), such as solar panels [1] and energy storage devices, balancing supply and demand becomes more challenging for a centralized controller. These challenges stem from the fact that supply (e.g., solar, wind, etc.)[2] and demand (e.g., electric vehicle charging) are more volatile. They also more frequently occur "behind the meter," meaning they are harder to measure. These conditions are further complicated since they are controlled by many different users, making them hard to manage [3]. Maintaining the balance between supply and demand is vital because imbalances shift the frequency of *alternating current* (AC) power, which in turn alters the behavior of devices connected to the power line. If these errors are not appropriately addressed, they can lead to physical system damage.

These challenges with DER have led to a recent surge of interest in peer-to-peer *transactive energy systems* (TES), as shown in Fig. 1. In TES, energy producers and consumers (known as "prosumers") submit offers to trade energy. If the trades in the system are not balanced—or if they exceed the safety constraints of the system—they can cause physical damage.

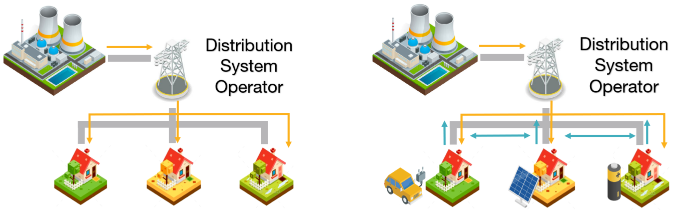

Fig. 1. DER Enables Transition from Top-down Energy Distribution to Peer-to-Peer Energy Transfers.

Safe trades can be incentivized by recording trades and fining prosumers for deviating from them. Providing this degree of oversight, however, requires knowledge about which prosumer has made a given offer. In turn, this type of information could violate privacy concerns if it is not carefully managed.

An example TES is shown in Fig. 2, where each node is associated with a participant in the local peer-to-peer energy trading market. This figure shows how a feeder consists of a number of nodes, some of which have the capability to sell energy. Each feeder is protected by an overcurrent relay at the junction of the common bus. The inset figure shows that a node in the network has different types of loads, some of which can be scheduled, making it possible for a consumer to bid in advance for those loads. A smart meter ensures proper billing per node.

A. Common Problems in Transactive Energy Systems

Using TES effectively in practice requires addressing the following problems (which we cover in this paper):

1) Ensuring the physical stability and safety of the grid apparatus, e.g., dynamically balancing supply and demand without violating line capacity constraints.
2) Ensuring that a peer-to-peer market operates in a trustworthy manner, even if some of the nodes are malicious.

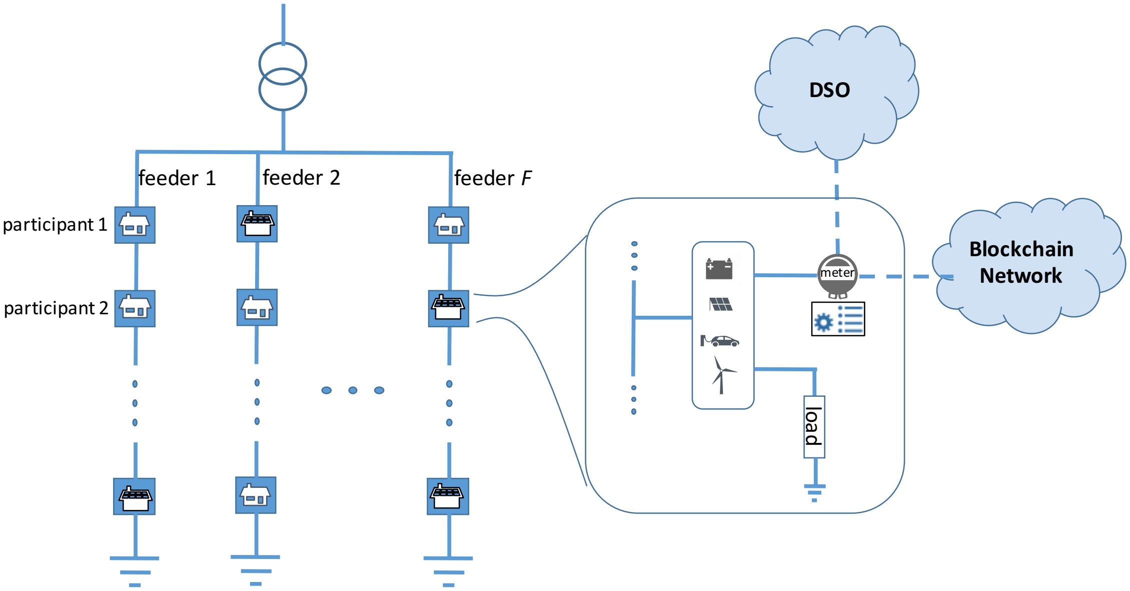

Fig. 2. **A Multi-feeder Microgrid Example of a Transactive Energy System.**

3) Ensuring privacy since TES disseminate information amongst participants to enable finding suitable trade partners.[1]

Our recent work [4] introduced *Privacy preserving Energy Transactions* (PETra), which is a blockchain-based solution that (1) enables trading energy futures in a secure and verifiable manner and (2) preserves prosumer privacy. These objectives were accomplished via a public distributed ledger for (1) and anonymized identifiers with a mixing service that prevents tracing the assets being traded back to the owner for (2). Our market mechanism was opportunistic, however, since each consumer looked at the available asks from producers and chose the one that fit the needs of the consumer the best, which was tedious and error-prone.

To address limitations with our prior work, this paper describes the structure, functionality, and performance of an *automated* matching system that maximizes the amount of energy traded within the local market. Moreover, we now consider system-wide safety constraints, whereas our prior work only considered constraints on individual prosumers. As in our prior work [4], we use the Ethereum blockchain platform to implement parts of our design via distributed ledgers. We chose Ethereum because it meets our requirements better than the alternatives. In particular proof-of-work consensus is better understood and has more formal analysis ([5], [6], [7]). There are alternatives such as Hyperledger Sawtooth [8] which uses a "Proof of Elapsed Time" consensus algorithm, however this relies on the Intel SGX hardware and we did not want to include that requirement in the system. Another alternative is Hyperledger Fabric [9] whose consensus is deterministic and relies on a known number of participants in the blockchain network and does not handle partitions well. We want a transactive energy market to continue operating despite partitions. To fully attain this goal we would require a special version of the consensus protocol that allowed two chains of transactions to be merged after the partition healed since trades must have happened on both sides. This is something to be explored in the future. Additionally we required a blockchain platform that is popular and well-documented. This approach is consistent with the recent trends in the research community and power industry focused on transactive energy markets [10], [11].

Disintermediation of trust is widely regarded as the primary feature of blockchain-based transaction systems [12]. Applying them in TES is appealing since they elegantly integrate the ability to immutably record the ownership and transfer of assets, with essential distributed computing services, such as Byzantine fault-tolerant consensus on the ledger state, as well as event chronology. The ability to establish consensus on state and ordering of events is important in the context of TES to detect trades that could destabilize the system.

B. Research Contributions

This paper makes the following contributions to research on transactive energy systems (TES):

- We co-design an automated matching mechanism and a decentralized transaction management platform that supports the energy trading workflow while ensuring prosumer privacy (i.e., their identity) and system safety

[1] In contrast, non-transactive smart metering systems require sharing prosumer information only with the *distribution system operator* (DSO), which eliminates/minimizes unauthorized privacy disclosures.

(e.g., satisfaction of line capacity constraints). This design problem is hard due to inherent conflicts between safety, privacy, and market efficiency.

- We enable the market controller to consider the effects of energy storage in batteries by enabling prosumers to specify multiple time intervals in which they could trade energy by explicitly indicating their flexibility.
- We describe the architecture and the protocol specification of our platform. Our solution combines the security and immutability of blockchain-based smart contracts with the efficiency of traditional computational platforms via a *hybrid solver* that is used to match energy trades.
- We present an experimental evaluation of our *Transaction Management Platform* (TMP) and the resulting market performance, with and without the availability of prosumer-owned battery storage. We consider total energy trade throughput as the market performance metric.

Paper Organization The remain of this paper is organized as follows: Section II describes our transactive microgrid model and reviews system requirements; Section IV formalizes the energy trading problem used as a case study throughout this paper; Section V examines our hybrid approach to solve the energy trading problem efficiently despite our use of a decentralized computing platform; Section VI explores the structure and functionality of a TMP we developed to provide the energy trading and market clearing functionality described in earlier sections. Section VII analyzes the results of experiments we conducted to evaluate the performance of our system; and Section VIII presents concluding remarks and lessons learned from our work.

II. System Model

We consider a microgrid with a set of feeders arranged in a radial topology. Although the methods presented in this paper are extensible to general tree topologies involving branching, we apply a radial topology to simplify checking the load flow constraints. A feeder has a fixed set of nodes, each representing a residential load or a combination of load and *distributed energy resources* (DERs), such as rooftop solar and batteries, as shown in Fig. 2. Each node is associated with a participant in the local peer-to-peer energy trading market.

Fig. 2 shows that a *distribution system operator* (DSO) also participates in the market. It may use this market to incentivize timed energy production within the microgrid to stabilize the grid and promote ancillary services [13], such as those outlined in Section I. Moreover, the DSO supplies residual demand not met through the local market.

Participants in our system model settle trades in advance, which allows them to schedule their power transfers into the local distribution system. There are typically three phases in these operations: (1) discovery of compatible offers, (2) matching of buying offers to selling offers (either by each prosumer individually or by an automated matching algorithm), and (3) performing the energy transaction and financial transaction.

A. System Requirements

Below we describe the requirements we addressed when building a decentralized *Transaction Management Platform* (TMP) that supports the workflow across the microgrid described above.

1) Communication Fabric: The first requirement is the existence of an appropriate communication and messaging architecture. The TMP must collect participants' offers and make them available to buyers and sellers. Moreover, the market algorithm must communicate clearing prices and buyer-seller matchings.

To meet the operational and safety requirements described below, these messages must be delivered reliably under strict timing constraints derived from the deadline by which a trade must clear. Moreover, the TMP must be capable of handling high volumes of micro-transactions anticipated in peer-to-peer trading scenarios. Finally, the communication fabric must support confidentiality, integrity, and non-repudiation of transactional data.

2) Operational Safety and Stability: Trading activity should not compromise the stability of the physical system operation. For instance, capacity constraints along any feeder should be respected, e.g., each feeder is rated for a maximal power capacity.[2] Local energy trade settlements should therefore ensure the instantaneous power flows stemming from power production and consumption never violate safety constraints.

3) Market Security and Efficiency: The TMP must include provisions for ensuring the protection of prosumer interests, as well as those of the DSO. Prosumer interests include being billed correctly based on energy prices set by the market and the measurements made by smart meters. In the context of microgrids connected to the broader power grid, the system should match supply and demand as closely as possible, while respecting safety constraints. In particular, the TMP should aim to maximize the amount of energy traded.

4) Privacy: Information like the amount of energy produced, consumed, bought, or sold by any prosumer should be available only to the DSO.[3] The owners of the bids and asks should remain anonymous to other participants. It should not be possible to infer a participant's energy usage patterns and personal information, such as financial standing, from their trading activity. In particular, inference of energy usage patterns can be exploited by inferring the presence or absence of a person in their home.

III. State-of-the-Art and Related Work Review

This section presents an overview of the state-of-the-art and compares our approach with related work on *transactive energy systems* (TES).

[2] Physically, the capacity limit of a feeder can be enforced using an overcurrent protection unit that limits the total current flowing through the feeder.

[3] This paper focuses on building a trading system, so we do not explicitly address the problem of billing. Multiple billing approaches can be implemented using blockchain technology, however, some providing a high degree of privacy, which is the focus of our future work.

A. Existing Implementations

The Brooklyn Microgrid project is a practical implementation of a TES [14]. This project was initially evaluated in terms of its ability to meet requirements deemed necessary for an efficient microgrid energy market. The first requirement noted by the authors (which they call the“microgrid setup”) defines who the participants are and their respective capabilities, access control, and assumptions regarding their behavior (honesty, rationality, etc.). This setup also defines the system objectives and the form of energy to trade, as well as the physical transfer mechanism, whether it be via the bulk power grid or an internal microgrid.

The second requirement the authors describe is a “grid connection,” which is the connection from a microgrid to the bulk power grid. This component serves the same purpose as the specification of the transfer mechanism in the first requirement above. The authors’ third requirement identified is the “information system,” which hosts the market and provides the communication framework for participants using a private blockchain implementation built upon Tendermint [15], which is a platform for replicating applications securely and consistently on many machines. This platform provides a variant of the *Practical Byzantine Fault Tolerance* (PBFT) consensus mechanism and a general-purpose application interface [16].

The information system outlined above hosts the fourth required component called the “market mechanism.” This mechanism defines how transactions are made, including payment and allocation rules. These rules ensure that the trades do not violate grid power constraints.

The market mechanism defines the fifth requirement, known as the “pricing mechanism”, which determines the cost of energy to exchange. This pricing mechanism is implemented using Ethereum “smart contracts” [17]. In this context, a smart contract is a custom program that operates on the data stored on a distributed ledger. It can be used to establish agreements between participants by ensuring some computation occurs when conditions specified in the program are true.

The final two required components defined by the Brooklyn Microgrid project are an automated trader for the participants and the regulations in which the TES is deployed. Despite the authors’ thorough description of these market components, they did not assess the costs associated with using a blockchain-based ledger as their information system, which could limit *efficiency* of the system. They also rely on *privacy* associated with using public keys, though keys in blockchain-based systems can be associated to owners through transaction-graph analysis [18].

The Brooklyn Microgrid system ensures *safety* and stability via the connection to the bulk power grid. Their approach, however, does not allow independent operation since this would require time synchronized action, which was not part of their design. Their system’s *resilience* is thus limited.

The Brooklyn Microgrid project utilized a smart contract to implement the market mechanisms, as outlined above. Smart contracts, however, often suffer from vulnerabilities that are hard to correct due to the nature of distributed ledgers. For example, Newman [19] analyzed 19,366 smart contracts and found that 8,833 contracts had one or more security issues.

Errors in smart contracts can result in devastating security incidents. For example, in the “DAO attack” $50 million in cryptocurrency was stolen [20] and in the multi-signature Parity Wallet library hack $280 million in cryptocurrency was lost. These vulnerabilities highlight that if such approaches are used, care is needed to ensure contract correctness due to the significant impact on *safety*, *security*, *resilience*, and *reliability*.

Vieira, the author of [21], implements a micromarket setup based on the Brooklyn Microgrid project where they compare the costs of traditional energy systems to a distributed ledger-based microgrid market. Vieira’s paper examines two auction mechanisms: (1) a continuous double auction and (2) a uniform-price double-sided auction. Vieira concludes that the continuous double auction could be implemented with Ethereum, though it is subject to expensive transaction costs. A Uniform-price double-sided auction was too complex using Ethereum, so it was performed on a centralized server.

B. Understanding the Gaps

Implementing a *Transaction Management Platform* (TMP) requires a communication architecture, as well as trading mechanisms that provide the capability to match bids and asks. Blockchain-based solutions have the potential to enable large-scale energy trading based on distributed consensus systems. However, popular blockchain solutions, such as Bitcoin [22] and Ethereum [23], possess design limitations that prevent their direct application to matching energy trades. These limitations stem from the complexity of additional constraints and checks required, beyond the transactional integrity check provided by proof-of-work algorithms.

For example, Aitzhan and Svetinovic [24] implemented a proof-of-concept platform for decentralized smart grid energy trading using blockchains. Their system is based on proof-of-work consensus and they do not consider grid control, stability, or scalability. Moreover, their approach still incurs privacy problems since all transactions in these systems are public [25].

Most related work discussing privacy examines it in the context of smart meters. For example, McDaniel and McLaughlin [26] discuss privacy concerns of energy usage profiling, which smart grids potentially enable. Efthymiou and Kalogridis [27] describe a method for securely anonymizing frequent electrical metering data sent by a smart meter via a third-party escrow mechanism.

Tan et al. [28] study privacy in a smart metering system from an information-theoretic perspective in the presence of energy harvesting and storage units. Their results show that energy harvesting provides increased privacy by diversifying the energy source, while a storage device can be used to increase both energy efficiency and privacy. Transaction data, however, provides more fine-grained information than smart meter usage patterns [29]. Majumder et al. [30] present an iterative double auction trading mechanism that preserves participants' privacy. However, this privacy property pertains to the participants' utility function models, not their identities.

Existing energy trading markets, such as the European Energy Exchange [31] and project NOBEL in Spain, employ the double-auction market mechanism [32], which can preserve participant privacy. Typical exchange implementations, however, involve centralized database architectures that are prone to single points of failure.

Faqiry and Das [33] present an auction mechanism for maximizing social welfare of buyers and sellers if the supply is small. Their approach also provides some privacy, i.e., participants do not reveal their utility function. The objective function of the microgrid controller is to maximize the power sold. The social welfare objective function can simultaneously be maximized by constricting buyers' utility functions to be convex.

To make the trading fair, the latter part of Faqiry and Das's paper considers an approach that discards the privacy maintained during the first phase. Their work contains no mechanism to check whether the buyer can produce the power they claim they can supply, which may result in instability. The authors also briefly mentioned that a distributed algorithm could be applied to their auction mechanism, though they did not implement this approach.

In contrast, our work presented in this paper provides a distributed systems mechanism that considers the problem of a broader definition of privacy, safety, and protection from malicious actors as a combined problem.

IV. Overview of the Energy Trading Problem

This section formulates the problem of matching energy future bids with asks (i.e., matching offers to buy energy intended for delivery in the future with offers to sell energy) in the local energy trading market. Our formulation aims to promote market efficiency by maximizing the amount of energy futures traded within the microgrid, while satisfying microgrid safety requirements. We first introduce an initial problem formulation that solves the energy trading problem offline. We then describe the version of this problem that considers a stream of incoming offers, which are cleared periodically.

A. Safety Requirements

While trades are being cleared, we must consider *safety requirements*. At the distribution level, the amount of power that can be sent over a transmission line is typically limited by the thermal properties of the conductor and is physically enforced by protection equipment, such as overcurrent relays. In traditional power systems, these capacity constraints are enforced by deploying some combination of load and generation curtailment schemes, which effectively impose upper bounds on the amount of power consumed by each load and the amount of power injected into the network by each source. For dispatchable generation, these upper bounds are typically calculated by solving some variant of an optimal power flow problem [34].

The settings of a TES is fundamentally different from settings in which economic dispatch is appropriate, i.e., privately-owned DERs are not dispatchable by the DSO. We therefore implement line capacity safety constraints by formulating them as constraints in the trading problem described below.

B. Problem Formalization

We begin by introducing a notation for elements of the microgrid, which is summarized in Table I. We let $\mathcal{F}$ denote the set of feeders. For a feeder $f \in \mathcal{F}$, we let C_f^{ext} denote the maximum amount of power allowed to flow into or out of the feeder at any point in time. Similarly, we let C_f^{int} denote the maximum amount of power that can be consumed or produced within the feeder at any point in time.[4] We assume that time is divided into intervals of fixed length Δ and we refer to the tth interval as time interval t.

TABLE I
List of Symbols

Symbol	Description
	Microgrid
$\mathcal{F}$	set of feeders
C_f^{ext}	maximum net power consumption or net power production in feeder $f \in \mathcal{F}$
C_f^{int}	maximum total power consumption or total power production in feeder $f \in \mathcal{F}$
Δ	length of time intervals
T_{clear}	minimum number of time intervals between the finalization and delivery of a trade
	Offers
$\mathcal{S}_f$	set of selling offers from feeder $f \in \mathcal{F}$
$\mathcal{B}_f$	set of buying offers from feeder $f \in \mathcal{F}$
$\mathcal{S}$, $\mathcal{B}$	set of all selling and buying offers, resp.
$\mathcal{S}^{(t)}$, $\mathcal{B}^{(t)}$	set of selling and buying offers submitted by the end of time interval t, resp.
E_s, E_b	amount of energy to be sold or bought by offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
I_s, I_b	time intervals in which energy could be provided or consumed by offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
R_s, R_b	reservation prices of offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
$\mathcal{M}(s)$, $\mathcal{M}(b)$	set of offers that are matchable with offers s and b, resp.
$I(s,b)$	$I_s \cap I_b$
	Solution
$p_{s,b,t}$	amount of energy that should be provided by s to b in interval t
$\pi_{s,b,t}$	unit price for the energy provided by s to b in interval t
$\mathit{Feasible}(\mathcal{S},\mathcal{B})$	set of feasible solutions given sets of selling and buying offers $\mathcal{S}$ and $\mathcal{B}$
$\hat{p}_{s,b,t}$, $\hat{\pi}_{s,b,t}$	finalized trade values
	Implementation Parameters
L	prediction window used by prosumers when posting selling and buying offers ($\min(L) = 2$)
$\hat{\Delta}$	length of the time step used for simulating the real-interval of length Δ
Δ_s	periodicity of solver that matches offers

[4] In other words, limit C_f^{ext} is imposed on the net production and net consumption of all prosumers in feeder f, while limit C_f^{int} is imposed on the total production and total consumption.

The input of the energy trading problem is the set of buying and selling offers posted by the participants.[5] For feeder $f \in \mathcal{F}$, we let $\mathcal{S}_f$ and $\mathcal{B}_f$ denote the set of selling and buying offers posted by participants located in that feeder, respectively.[6] A selling offer $s \in \mathcal{S}_f$ is a tuple (E_s, I_s, R_s), where

- E_s is the amount of energy to be sold,
- I_s is the set of time intervals in which the energy could be provided,
- R_s is the reservation price, i.e., lowest unit price for which the participant is willing to sell energy.

Similarly, a buying offer $b \in \mathcal{B}_f$ is a tuple (E_b, I_b, R_b), where the values pertain to consuming/buying energy instead of producing/selling, and R_b is the highest price that the participant is willing to pay. For convenience, we also let $\mathcal{S}$ and $\mathcal{B}$ denote the set of all buying and selling offers (i.e., we let $\mathcal{S} = \cup_{f \in \mathcal{F}} \mathcal{S}_f$ and $\mathcal{B} = \cup_{f \in \mathcal{F}} \mathcal{B}_f$).

We say that a pair of selling and buying offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$ is *matchable* if

$$R_s \leq R_b \tag{1}$$

$$I_s \cap I_b \neq \emptyset. \tag{2}$$

In other words, a pair of offers is matchable if there exists a price that both participants would accept and a time interval in which the seller and buyer could provide and consume energy. For a given selling offer $s \in \mathcal{S}$, we let the set of buying offers that are matchable with s be denoted by $\mathcal{M}(s)$. Similarly, we let the set of selling offers that are matchable with a buying offer b be denoted by $\mathcal{M}(b)$ and we let $I(s, b) = I_s \cap I_b$.

A solution to the energy trading problem is a pair of vectors $(\boldsymbol{p}, \boldsymbol{\pi})$, where

- $p_{s,b,t}$ is a non-negative amount of power that should be provided by the seller $s \in \mathcal{S}$ and consumed by the buyer $b \in \mathcal{M}(s)$ in time interval $t \in I(s, b)$.[7]
- $\pi_{s,b,t}$ is the unit price for the energy provided by seller $s \in \mathcal{S}$ to buyer $b \in \mathcal{M}(s)$ in time interval $t \in I(s, b)$.

A pair of vectors $(\boldsymbol{p}, \boldsymbol{\pi})$ is a feasible solution to the energy trading problem if it satisfies the following constraints:

- The amount of energy sold or bought from each offer is at most the amount of energy offered:

$$\forall s \in \mathcal{S} : \sum_{b \in \mathcal{M}(s)} \sum_{t \in I(s,b)} p_{s,b,t} \cdot \Delta \leq E_s \tag{3}$$

$$\forall b \in \mathcal{B} : \sum_{s \in \mathcal{M}(b)} \sum_{t \in I(s,b)} p_{s,b,t} \cdot \Delta \leq E_b \tag{4}$$

- The amount of power flowing into or out of each feeder is below the safety limit in all time intervals:

$$\forall f \in \mathcal{F}, t : \left(\sum_{s \in \mathcal{S}_f} \sum_{b \in \mathcal{B}} p_{s,b,t} \right) - \left(\sum_{b \in \mathcal{B}_f} \sum_{s \in \mathcal{S}} p_{s,b,t} \right) \leq C_f^{ext} \tag{5}$$

$$\forall f \in \mathcal{F}, t : \left(\sum_{s \in \mathcal{S}_f} \sum_{b \in \mathcal{B}} p_{s,b,t} \right) - \left(\sum_{b \in \mathcal{B}_f} \sum_{s \in \mathcal{S}} p_{s,b,t} \right) \geq -C_f^{ext} \tag{6}$$

- The amount of energy consumed and produced within each feeder is below the safety limit in all time intervals:

$$\forall f \in \mathcal{F}, t : \sum_{b \in \mathcal{B}_f} \sum_{s \in \mathcal{S}} p_{s,b,t} \leq C_f^{int} \tag{7}$$

$$\forall f \in \mathcal{F}, t : \sum_{s \in \mathcal{S}_f} \sum_{b \in \mathcal{B}} p_{s,b,t} \leq C_f^{int} \tag{8}$$

- The unit prices are between the reservation prices of the seller and buyer:

$$\forall s \in \mathcal{S}, b \in \mathcal{M}(s), t \in I(s, b) : R_s \leq \pi_{s,b,t} \leq R_b \tag{9}$$

The objective of the energy trading problem is to maximize the amount of energy traded. Formally, an optimal solution to the energy trading problem is

$$\max_{(\boldsymbol{p}, \boldsymbol{\pi}) \in \textit{Feasible}(\mathcal{S}, \mathcal{B})} \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{M}(s)} \sum_{t \in I(s,b)} p_{s,b,t} , \tag{10}$$

where *Feasible*$(\mathcal{S}, \mathcal{B})$ is the set of feasible solutions given selling and buying offers $\mathcal{S}$ and $\mathcal{B}$ (i.e., set of solutions satisfying Equations (3) to (9) with $\mathcal{S}$ and $\mathcal{B}$).

C. Advanced Problem Formulation

Our basic problem formulation described in Section IV-B assumed that (1) all buying and selling offers $\mathcal{B}$ and $\mathcal{S}$ are available at once and (2) the market is cleared in one take. In practice, however, prosumers and the DSO may continuously submit new offers as their predictions, their physical state, and the market conditions change over time. As the set of submitted offers grows, the optimal solution to the energy trading problem may change, and the optimal value of each $p_{s,b,t}$ may vary.

While each change can increase the amount of energy traded, the *trade values* $p_{s,b,t}$ and $\pi_{s,b,t}$ must be *finalized* at some point in time. At the very latest, values for interval t must be finalized by the end of interval $t-1$ or else participants would have no chance of actually delivering the trade. We therefore extend the energy trading problem to consider a growing set of offers and a time constraint for finalizing trades. Our approach finalizes a minimum set of trades in each interval, which maximizes efficiency while providing safety.

We assume that all trades for time interval t (i.e., all values $p_{s,b,t}$ and $\pi_{s,b,t}$) must be finalized by the end of time

[5] Participants may include both prosumers and the DSO. The DSO can shape load and trade energy futures by participating in the energy market in the same way as prosumers.

[6] If the DSO wants to participate in this energy trading market, it can be assigned to a "dummy" feeder in the problem formulation.

[7] We require the both the seller and buyer to produce a constant level of power during the time interval.

interval $t - T_{clear}$, where T_{clear} is a positive integer constant, which is set by the operator. Preventing "last-minute" changes is crucial for safety and fairness since it allows both the DSO and the prosumers to prepare for delivering (or consuming) the right amount of power. In practice, the value of T_{clear} must be chosen by taking into account both physical constraints (e.g., how long it takes to turn on a generator) and communication delay (e.g., some participants may be delayed in learning of a trade due to network disruptions).

We let $\hat{p}_{s,b,t}$ and $\hat{\pi}_{s,b,t}$ denote the finalized trade values. Likewise, we let $\mathcal{B}^{(t)}$ and $\mathcal{S}^{(t)}$ denote the set of buying and selling offers that participants have submitted by the end of time interval t. The system then takes the following steps at the end of each time interval t:

- Find an optimal solution $(\boldsymbol{p}^*, \boldsymbol{\pi}^*)$ to the extended energy trading problem:

$$\max_{(\boldsymbol{p},\boldsymbol{\pi}) \in \, \mathit{Feasible}(\mathcal{S}^{(t)},\, \mathcal{B}^{(t)})} \sum_{s \in \mathcal{S}^{(t)}} \sum_{b \in \mathcal{M}(s)} \sum_{\tau \in I(s,b)} p_{s,b,\tau} \tag{11}$$

  subject to

$$\forall \tau < t + T_{clear}\colon \; p_{s,b,\tau} = \hat{p}_{s,b,\tau} \tag{12}$$

$$\pi_{s,b,\tau} = \hat{\pi}_{s,b,\tau} \tag{13}$$

- Finalize trade values for time interval $t + T_{clear}$ based on the optimal solution $(\boldsymbol{p}^*, \boldsymbol{\pi}^*)$:

$$\hat{p}_{s,b,t+T_{clear}} := p^*_{s,b,t+T_{clear}} \tag{14}$$

$$\hat{\pi}_{s,b,t+T_{clear}} := \pi^*_{s,b,t+T_{clear}} \tag{15}$$

By performing the steps outlined above at the end of each time interval, trades are always cleared based on as much information as possible (i.e., considering as many offers as possible) without violating any safety or timing constraints.

V. Our Solution Approach to the Energy Trading Problem

This section presents our hybrid approach for solving the energy trading problem on a decentralized computing platform. This hybrid approach combines the auditability and trustworthiness of Ethereum blockchain-based smart contracts with the efficiency of more traditional computing platforms. We first show how to solve the problem by formulating it as a linear program and then describe the computation and verification performed by the computational nodes and the smart contract in a decentralized microgrid.

A. Linear-Programming Solution

We solve the basic energy trading problem efficiently by formulating it as a linear program. First, we create real-valued variables $p_{s,b,t}$ and $\pi_{s,b,t}$ for each $s \in S, b \in \mathcal{M}(s), t \in I(s,b)$. We then reformulate the matching problem as a linear program:

$$\max_{\boldsymbol{p},\boldsymbol{\pi}} \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{M}(s)} \sum_{t \in I(s,b)} p_{s,b,t} \tag{16}$$

subject to Equations (3) to (9) and

$$\boldsymbol{p} \geq \boldsymbol{0} \text{ and } \boldsymbol{\pi} \geq \boldsymbol{0}. \tag{17}$$

The extended energy trading problem introduced in Section IV-C can similarly be reformulated as a linear program by considering $\mathcal{S}^{(t)}$, $\mathcal{B}^{(t)}$, $\hat{\boldsymbol{p}}$, $\hat{\boldsymbol{\pi}}$, and the additional constraints.

B. Hybrid Solver Implementation

Although solving linear programs is not computationally hard, it can be challenging in resource-constrained computing environments with a large number of variables and constraints. Moreover, computation is relatively expensive on blockchain-based distributed platforms, so solving the energy trading problem using a blockchain-based smart contract does not scale in practice.[8] We therefore apply a *hybrid approach* that combines the trustworthiness of blockchain-based smart contracts with the efficiency of more traditional computational platforms.

Our hybrid approach uses a high-performance computer to solve the computationally expensive linear program *off-blockchain* and then applies a smart contract to record the solution *on the blockchain*. Implementing this hybrid approach securely and reliably required us to address the following issues:

- Computations performed off-blockchain do not satisfy the auditability and security requirements that smart contracts do. The results of any off-blockchain computations must therefore be verified in some way by the smart contract before recording them on the blockchain.
- Due to network disruptions and other errors (including deliberate denial-of-service attacks), the off-blockchain solver may not provide the smart contract with a solution on time (i.e., before trades should be finalized). The smart contract must therefore be able to proceed without an up-to-date solution.
- To enhance reliability, the smart contract should accept solutions from multiple off-blockchain sources, though these sources may provide different solutions. The smart contract must therefore be able to choose from multiple solutions, some of which may come from compromised computers.

1) A Blockchain-based Smart Contract: We implement a smart contract that can (1) verify whether a solution $(\boldsymbol{p}, \boldsymbol{\pi})$ is feasible and (2) compute the value of the objective function for a feasible solution. Compared to finding an optimal solution, these operations are computationally inexpensive and can thus be performed on a blockchain-based decentralized platform. Our smart contract uses these capabilities to provide the following functionality:

- Solutions may be submitted to the contract at any time. The contract verifies the feasibility of each submitted solution and if the solution is feasible the value of the objective function is then computed. The smart contract always keeps track of the best feasible solution submitted so far, which we call the *candidate solution*.

[8] Solidity (the preferred high-level language for Ethereum) currently lacks built-in support for certain features (such as floating-point arithimetic) that would facilitate the implementation of a linear programming solver.

- At the end of each time interval t, the smart contract finalizes the trade values for interval $t + T_{clear}$ based on the candidate solution.[9]

This smart contract functionality achieves a high level of security and reliability. In particular, an adversary cannot force the contract to finalize trades based on an unsafe (i.e., infeasible) solution since such a solution would be rejected. Likewise, an adversary cannot force the contract to choose an inferior solution instead of a superior one. The only action available to an adversary, therefore, is proposing a superior feasible solution, which would actually improve energy trading in the microgrid.

The contract is also reliable and can tolerate temporary disruptions in the solver or the communication network. In fact, any solution $(\boldsymbol{p}, \boldsymbol{\pi})$ that is feasible for sets $\mathcal{S}$ and $\mathcal{B}$ is also feasible for supersets $\mathcal{S}' \supseteq \mathcal{S}$ and $\mathcal{B}' \supseteq \mathcal{B}$. As the sets of offers only grow over time, the contract can use a candidate solution submitted during time interval t to finalize trades in any subsequent time interval $\tau > t$.

Without receiving new solutions, the difference between the amount of finalized trades and the optimum only increases gradually. In particular, the earlier candidate solution can specify trades for any future time interval. The difference is therefore due only to the offers that have been posted since the solution was found and submitted.

2) An Off-blockchain Solver: We complement the smart contract with an efficient linear programming solver (e.g., CPLEX [35]). This solver can be run off-blockchain on any capable computer or on multiple computers for reliability. The solver runs periodically to find a solution to the energy trading problem based on the latest set of offers posted.

After the solver finds a solution, it is submitted to the smart contract in a blockchain transaction. If new offers have been posted since the solver started working on the solution, the contract will still consider the solution as feasible. This capability stems from the fact that any feasible solution for sets $\mathcal{S}$ and $\mathcal{B}$ are also feasible for supersets $\mathcal{S}' \supseteq \mathcal{S}$ and $\mathcal{B}' \supseteq \mathcal{B}$.

From the solver's perspective, there are many advantages of submitting multiple solutions to the smart contract for the same problem. For example, this design allows the linear programming solver to run as an "anytime algorithm." Moreover, multiple—potentially untrusted—entities can try to solve the problem and submit solutions since the smart contract always chooses the best feasible one, which is important in microgrids where a trusted third-party may no always be present. In such settings, prosumers can volunteer and provide solutions to the energy trading problem.[10] We therefore enable finding solutions in an efficient and flexible manner, while reaping smart contract benefits, such as auditability and trustworthiness.

VI. Structure and Functionality of Our Blockchain-based Energy Trading System

This section examines the structure and functionality of a system we developed to provide the energy trading and market clearing capabilities described in prior sections. We first summarize the run-time platform and then describe of the messages exchanged between these components in the trading workflow.

A. Overview of the RIAPS Platform

In any system that relies on distributed real-time and embedded computing, a key ingredient is the computing platform that provides the foundations for all algorithms, isolates the hardware details from the algorithms, and provides essential mechanisms for resource management, fault tolerance, and security. The work presented in this paper is based on the 'Resilient Information Architecture Platform for Smart Grid' (RIAPS) [36], which supports scalable, dependable, and secure dissemination of information between actors. Each actor in RIAPS is a composition of several libraries that provide the following capabilities:

- a component model that provides a concurrent model of computation for building distributed real-time applications,
- a messaging framework for facilitating interactions among actors,
- a resource-management framework for controlling the use of computational resources,
- a fault-management framework for detecting and mitigating faults in all layers of the system,
- a security framework to protect the confidentiality, integrity, and availability of system under cyber-attacks,
- a fault tolerant time synchronization service,
- a discovery framework for establishing the network of interacting actors in an application, and
- a deployment and management framework for administering and coordinating the distributed applications from a control room.

B. Energy Trading System Architecture

Figure 3 shows the architecture of our energy trading system. Our implementation uses the Ethereum blockchain

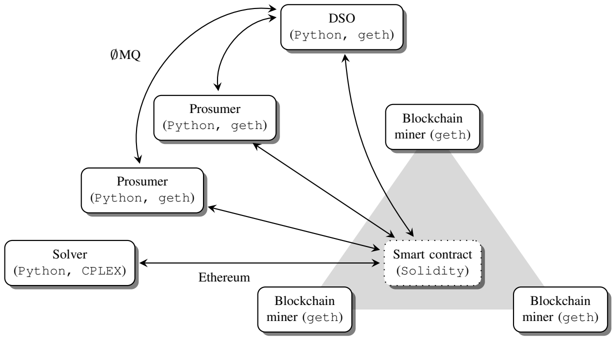

Fig. 3. Components of Our Blockchain-based Energy Trading System.

[9] If no solution has been submitted to the contract thus far (which may occur right after the trading system has been launched) $\boldsymbol{p} = \boldsymbol{0}$ may be used as a candidate solution.

[10] Although each prosumer will try to submit a solution that favors itself, the submitted solution must still be superior with respect to the optimization objective, which roughly corresponds to social utility. Hence, prosumers are incentivized to improve social utility by submitting a superior solutions that favors the prosumer, which we analyze in future work.

middleware as the decentralized computation platform running smart contracts. Other components interact with the blockchain network using the `geth` Ethereum client.

The smart contract is implemented in Solidity (which is a high-level language for programming Ethereum) and is executed by a network of `geth` mining nodes. The specific platforms and tools used in our implementation are shown in parentheses and as arrow labels in Figure 3. All components are written in Python and communicate with each other using ZeroMQ, which is high-performance asynchronous messaging middleware supported by the RIAPS platform.

C. Providing Privacy

To protect prosumer privacy, anonymous addresses are used when interacting with the blockchain (e.g., posting offers). By periodically generating new anonymous addresses at random, other participants are prevented from linking the anonymous addresses to their actual identities [4]. This approach ensures that trading activities are private. In contrast with our prior work [4], the addresses used in the workflow described in Section VI-D are not completely anonymous. Since the blockchain-based smart contract must check feeder-level safety constraints, each anonymous address must be linked to a feeder. An anonymous address can thus only hide the prosumer's identity, but not its feeder. This design is a manifestation of the trade-off between safety and privacy.[11]

Ironically, anonymous addresses pose a further threat to safety since participants can generate anonymous addresses at almost no cost, which enables them to post selling and buying offers for large amounts of energy, without any intention of delivering and without facing any repercussions. A malicious or faulty participant could easily destabilize the grid with this form of reckless trading. Consequently, the amount of energy that may be traded by anonymous addresses belonging to a participant must be limited.

To enforce this constraint, we employ the concept of energy assets, initially introduced in [4] and used in the trading workflow described in Section VI-D. An energy asset represents a permission to sell or buy a specific amount of energy in a specific set of time intervals. Each prosumer wanting to transfer assets to an anonymous address must first ask the DSO, which then checks whether it is safe to give permission to the participant. If so, the DSO records this transfer on the blockchain.

When a participant subsequently posts an offer from the anonymous address, the smart contract can check whether the address has the assets required for the offer. Only the DSO can link the anonymous address to the participant. These energy assets thus enforce participant-level constraints without violating privacy.

D. Energy Trading Workflow

Fig. 4 depicts the energy trading workflow.

Solid lines in this figure represent messages, including smart-contract function calls (i.e., blockchain transactions), whereas dashed lines represent smart-contract events. To simplify the presentation, the figure shows only one prosumer and only a single message of each type. In practice, a large number of prosumers may interact with the DSO and the smart contract, and each of them may exchange multiple messages.

The trading workflow includes the following messages:

- `withdrawAssets(anonAddress, energy, intervals, amount)`, which is a message sent by a prosumer to the DSO, asking the DSO to transfer energy and/or financial assets from the prosumer's account at the DSO to an anonymous address. Before sending this message, the prosumer first generates a random anonymous address to protect its privacy. This message specifies the assets the prosumer wishes to withdraw (i.e., amount of energy and time intervals) and the anonymous address to which the DSO should transfer them. The prosumer may send this message long before actually engaging in trading, so the DSO need not be online continuously.
- `failedWithdrawal(anonAddress, msg)`, which is a message sent by the DSO to the prosumer notifying the prosumer that the requested assets cannot be withdrawn, e.g., due to energy safety requirements or insufficient funds.
- `addEnergy(anonAddress, energy, intervals)`, `addFinancialBalance(anonAddress, amount)`, which are smart contract functions called by the DSO (i.e., transactions recorded on the blockchain) in response to the prosumer's request used to create energy and financial assets on the blockchain and transfer them to an anonymous address. Before recording this transaction, the DSO must first verify that enabling the prosumer to trade these assets does not violate any safety constraints and that the anonymous address is linked to the correct feeder.
- `AssetAdded(anonAddress, energy, intervals)`, `FinancialAdded(anonAddress, amount)`, which are broadcast messages emitted by the smart contract (i.e., events logged on the blockchain) notifying the prosumer that the requested assets have been transferred to the anonymous address.
- `postOffer(energy, intervals, price)`, which is a smart-contract function called by a prosumer (from its anonymous address), publicly posting an energy bid or ask.
- `OfferPosted(offerID, energy, intervals, price)`, which is a message broadcast by the smart contract notifying solvers that an offer was posted.
- `submitSolution(powers, prices)`, which is a smart-contract function called by a solver submitting a new solution for the energy trading problem.
- `SolutionFinalized(powers, prices)`, which is a message broadcast by the smart contract notifying both prosumers and solvers of energy trades that have been finalized.

[11] Actually, participants can remain anonymous among a class of feeders with same number of participants and identical safety constraints.

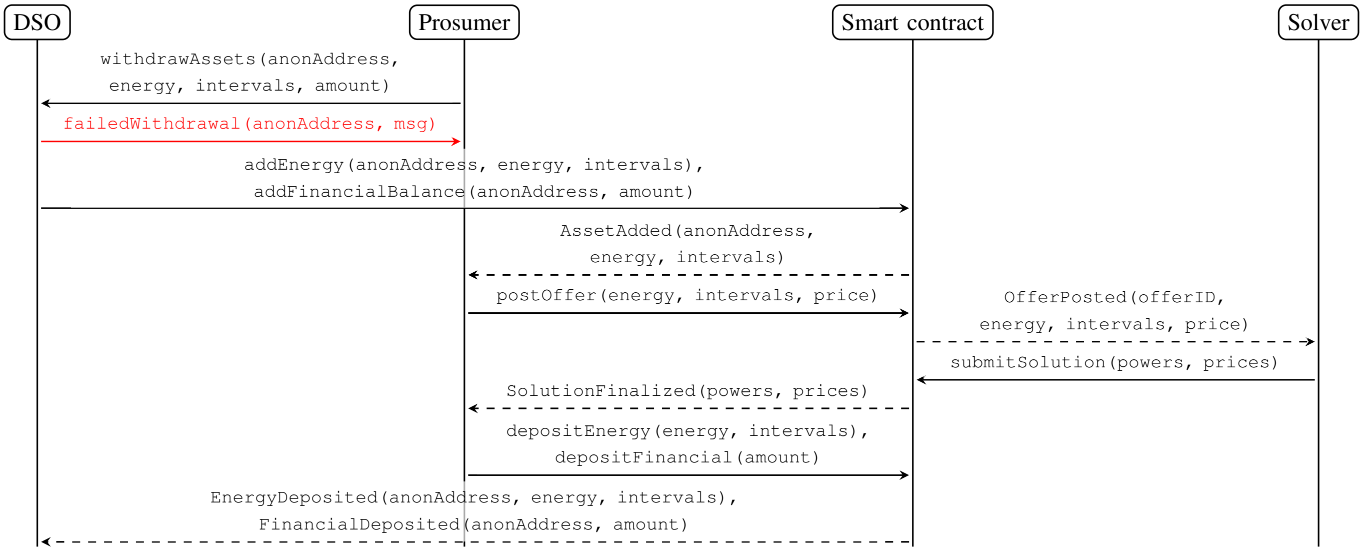
DSO
Prosumer
Smart contract
Solver
withdrawAssets(anonAddress, energy, intervals, amount)
failedWithdrawal(anonAddress, msg)
addEnergy(anonAddress, energy, intervals), addFinancialBalance(anonAddress, amount)
AssetAdded(anonAddress, energy, intervals)
postOffer(energy, intervals, price)
OfferPosted(offerID, energy, intervals, price)
submitSolution(powers, prices)
SolutionFinalized(powers, prices)
depositEnergy(energy, intervals), depositFinancial(amount)
EnergyDeposited(anonAddress, energy, intervals), FinancialDeposited(anonAddress, amount)

Fig. 4. Sequence Diagram of the Trading Workflow.

- `depositEnergy(energy, intervals)`, `depositFinancial(amount)`, which are smart-contract functions called by a prosumer to deposit energy and financial assets to the prosumer's account. To protect privacy, these calls do not specify the prosumer, so the DSO must keep track of which prosumer uses which anonymous address.
- `EnergyDeposited(anonAddress, energy, intervals)`, `FinancialDeposited(anonAddress, amount)`, which are messages broadcast by the smart contract notifying the DSO that assets have been deposited from anonymous address. These messages in turn triggers the transfer of the designated assets to the prosumer's account at the DSO.

E. Implementation Considerations

1) Parameters: The system of prosumers, solvers, DSO, and smart contract operates mostly asynchronously. They all operate as independent processes running on remote nodes, with their own time bases. In particular, the solver can operate as a periodic process (with a period Δ_s), waiting on information from the smart contract about all the offers that have been posted in the prior period. The only synchronous communication occurs between prosumers and the DSO.

Prosumers can also operate as periodic processes, submitting their offers and bids to the smart contract. In practice, prosumers are synchronized with real wall-clock time, making their bids and asks known for future intervals, depending upon the time at which they post their bids/asks and how far in the future they can predict their usage or operation. We make their prediction window L a parameter of the system.

The value of the predication window parameter is at least 2, because prosumers must make a bid/ask for at least the next interval (we count the current interval in L). A larger value of this prediction window increases the risk of uncertainty for the prosumer since they are expected to fulfill their bid or ask.[12] During the experiments presented in Section VII, we simulated Δ as $\hat{\Delta}$ to accelerate the process.[13] These parameters are described in Table I.

2) Speed and Synchronization Considerations: A relevant problem for TMP is deciding how fast it can run and ensuring that trades for the next interval can clear before the T_{clear} parameter, which has a minimum bound of 1. Our trading system assumes that regular network communication mechanisms are fast. Communication with the smart contract, however, is limited by the block mining rate and multiple messages must be exchanged in each interval (see Fig. 4).

Given these constraints, miners need to work fast enough to mine a few blocks in each time interval. In a closed environment, this goal can be achieved by reducing the difficulty of the cryptographic puzzle solved for proof-of-work consensus. In our system we can clear transactions much faster than one 15-minute time interval (see Fig. 11). To scale up performance in larger systems, the proof-of-work consensus may be replaced, e.g., by a proof-of-stake mechanism.

Another challenged addressed by our trading system was the synchronization between different agents. The RIAPS runtime platform provides high-precision time synchronization [37]. Even if only the *network time protocol* (NTP) is used as the synchronization mechanism, however, the system operates correctly for the following reasons:

- In practice, time intervals are relatively long (e.g., 15 minutes), compared to typical communication delays.
- Our smart contract ensures that the system always proceeds to the next time interval, though the accuracy of

[12] The size of the prediction window is part of the prosumer strategy, which is not explored in this paper since we focus on the implementation of the TMP.

[13] This value represents the amount of real time passed in the simulation before proceeding to the next interval, which allows us to accelerate the experiment since running the system slower would be easier.

this is limited by the mining rate.

- Our trading system can tolerate or discard out-of-order and delayed messages due to the event chronology implemented in the blockchain platform.

Nevertheless, prosumers should try to post their offers early within a time interval so that solvers include them in the solution for the current time interval. Conversely, solvers should wait for some time before starting to work so they can collect all (or at least most) of the offers posted in the interval.

VII. Experiment Testbed and Analysis of Results

This section analyzes the results of experiments performed to evaluate the performance of our energy trading system described in Section VI. The source code is available online at [38].

A. *Overview of the Microgrid Configuration*

The experiments presented in this section are based on a collection of load traces recorded by Siemens from a microgrid in Germany that contained 102 homes across 11 feeders (5 producers and 97 consumers). Figure 5 describes the feeder structure, the number of participants per feeder, and the feeder safety limits of this microgrid. Brown nodes in this diagram are feeder junctions, numbered 1 to 11 from top to bottom. Black nodes are the overcurrent relays, which ensure that the total power flowing in and out of the feeder is below 20 kW. Green nodes are the junction points for the producers (5) and red nodes are junction points for the consumers (97). There are 102 prosumers in total. We use $\Delta = 15$ minute intervals, resulting in a total of 96 intervals across the whole day.

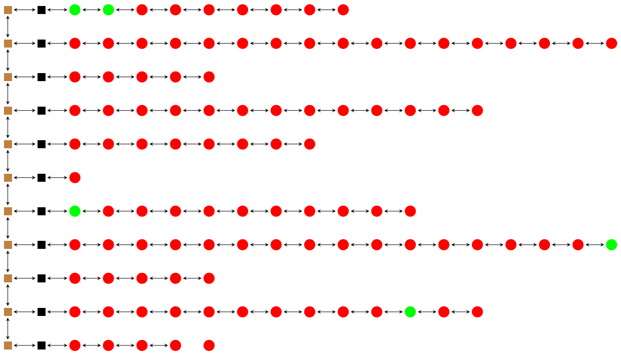

Fig. 5. Feeder Diagram.

Figure 6 shows the total production and consumption across this microgrid. The horizontal axis shows the starting time for each of the 96 intervals. Since the dataset does not include prices, we assume that reservation prices are uniform in our experiments and thus focus on studying the amount of energy traded and the performance of the system.

The graph in Figure 6 also shows the energy traded per interval without battery (Test A), and with battery Tests C and D (see Table II). The amount of energy traded can be lower than both supply and demand at the same time due to safety constraints. These constraints limit the amount of energy that can flow out of producers' feeders.

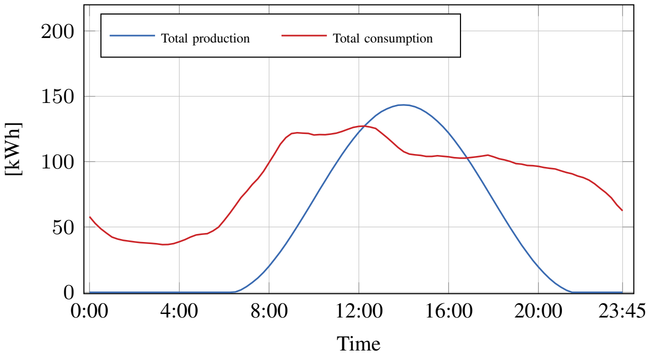

Fig. 6. Load Profile (i.e., Total Consumption) and Generation Profile (i.e., Total Production) in kWh per 15 Minute Intervals Aggregated Across the Microgrid.

A key contribution in our work is the ability to specify multiple time intervals for selling offers. We therefore extended the trace collected by Siemens to allow each producer to have a battery with a total capacity of 90 kWh. Producers can use their batteries to store energy generated within a time interval and make it available in future time intervals. The resulting offers always span a contiguous set of time intervals, so they can be specified by their starting time and length.

Figure 7 shows the time intervals for one producer. The red bars in this figure indicate the number of contiguous intervals for which the offer is valid. The total battery capacity is 90 kWh and we assume the battery charges at a rate of 10 kWh per interval. Producers are assumed to keep the battery available until the end of the test, which is the 95th interval, so the red bars taper off in consecutive intervals. Producers charge their batteries only when total consumption is less than total production, which happens just after 12:00 PM, as shown in Figure 6.

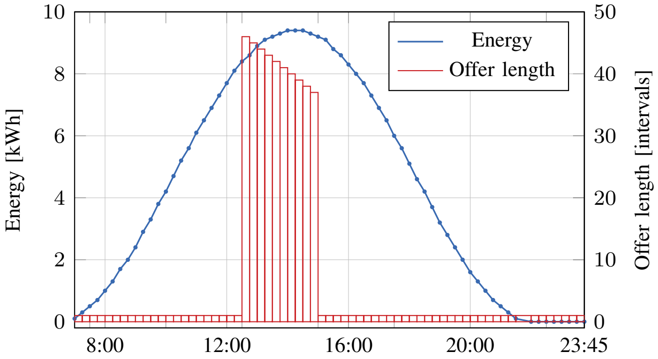

Fig. 7. Energy Offered in Each Time Interval by the First Prosumer of the First Feeder When Using a Battery (shown by the blue line).

B. *Experiment Testbed Structure and Functionality*

Our experiment testbed comprises a cluster of 31 BeagleBone Black (BBB) single-board computers, which are shown in Figure 8. These BBBs act as participants in the energy trading system and are configured as light clients because they are resource-constrained and thus not suitable for mining or acting as solvers. As a result, they can safely access the blockchain, but do not participate in the consensus process.

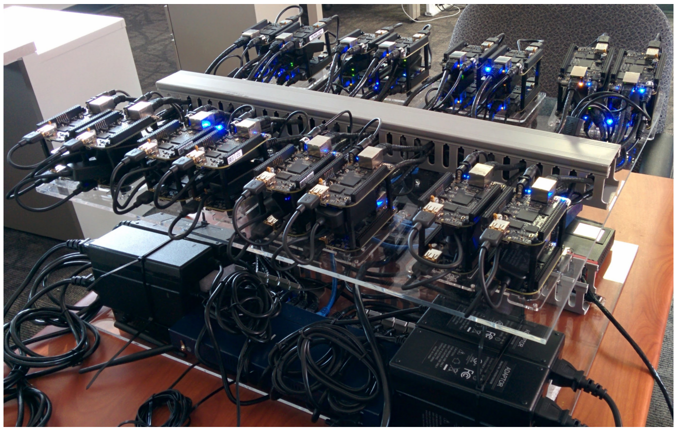

Fig. 8. Hardware Testbed.

In the dataset provided by Siemens, there are 95 consumers and 7 producers of power, as shown in Figure 5. These participants are divided between the BBBs in the cluster. The *Platform for Transactive IoT Blockchain Applications with Repeatable Testing* (PlaTIBART) environment [39] provided us with the necessary DevOps support.

The block mining is provided by external hardware, locally or in a cloud server. We had a single miner instance that maintained the blockchain and a single solver instance that used CPLEX [35] to solve the energy trading problem. This setup can easily be scaled up to add more miners and solvers if sufficient computational resources are available. The communication between the components was implemented using ZeroMQ, which a communication protocol supported by RIAPS.

C. Experiment Tests and Analysis of Results

Table II describes the specifics of the four categories of tests that we ran in our experiments. The tests vary the different implementation parameters (see Table I for symbol definitions and Section VI-E). This flexibility enabled us to study how changing these parameters affected the total amount of energy traded.

TABLE II
PARAMETER VALUES FOR EXPERIMENTS (SEE TABLE I FOR SYMBOL DEFINITIONS)

	A	B	C	D
Δ[m]	15	15	15	15
Δ_s [s]	5	5	5	5
$\hat{\Delta}$ [s]	120	120	120	120
L	2,3,5,7,10,13	2,3,7,10	5	13
Battery	no	yes	yes	yes
Figure	9	9	10,11,6,9	6,9

Figure 9 shows the total energy traded for different tests. We varied the prediction window (L) for the participants from 2 to 13, i.e., in each interval participants submitted offers starting from the next 1 to 12 intervals (the current interval is always counted in the prediction window). The experiment simulated the whole day from the first interval starting at 0:00 (12:00 AM) to the 95^{th} interval ending at 23:59 (11:59 PM).

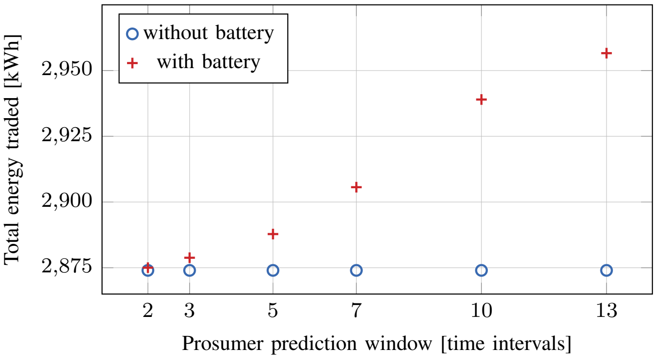

Fig. 9. Total Amount of Energy Traded in the Entire Microgrid (With and Without Batteries) for Various Prediction Window Lengths.

As expected, increasing the prediction window without batteries had no effect on the total amount energy traded. This result occurred because any production must be dispatched within one time interval. The solver therefore cannot optimize energy usage across multiple intervals, even if future offers are available.

If batteries were available in the system, however, the amount of energy traded increased since the solver matched offers across multiple time intervals at once. Trading increased with the prediction window due to the increased analysis space available to the solver. Fig. 6 shows the per interval trades for three of these cases (A, C, and D): without battery, with battery and $L = 5$, as well as with battery and $L = 13$.

Fig. 10 shows the energy matched per interval in test case C for the first prosumer of the first feeder. The stacked colors

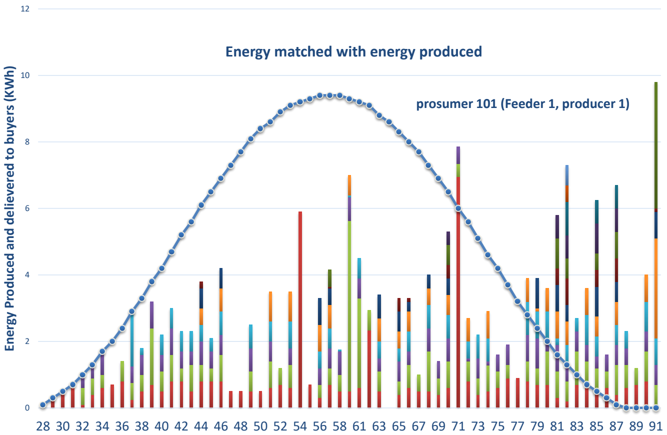

Fig. 10. Energy Generated in Each Interval (blue line) and Energy Traded to a Set of Consumers in Each Interval (vertical bars) for the First Prosumer of the First Feeder (Test case C).

show the different consumers that were matched with the prosumer in each interval (the same color across multiple intervals does not necessarily mean it is the same consumer). When the energy traded exceeded the generation, the excess was drawn from the battery.

Fig. 11 shows a histogram of the time between posting an offer and recording a trade on the blockchain that includes the offer (also in Test Case C). This time included the communication delay, the time to mine the blockchain, and the running time required to find a solution. The time length was always less than $\hat{\Delta}$, which was 120 seconds (see Table II). Since the solver ran periodically and received offers asynchronously, there were a few runs before a suitable match was found.

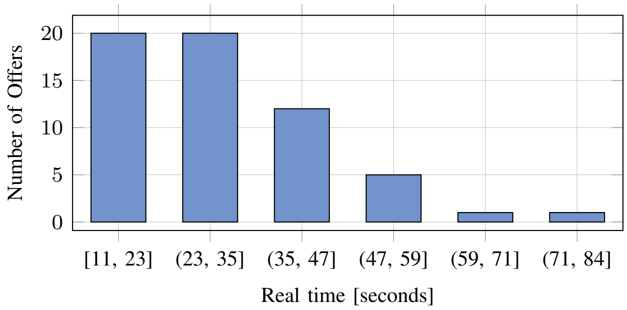

Fig. 11. Real Time in Seconds Between Posting an Offer and Recording a Trade that Includes the Offer (Test Case C).

VIII. Concluding Remarks

This paper describes the design and implementation of a *transaction management platform* (TMP) for transactive microgrids. Our solution enables prosumers to trade energy without threatening their privacy or compromising system safety. The Ethereum blockchain platform provides decentralized trust and consensus capabilities, which protect the transactive energy system from malicious actors. Our hybrid solver approach combines a validator based on Ethereum smart contracts with an external optimizer, which enables the platform to clear offers securely and efficiently. Likewise, the ability to trade across multiple time intervals enables participants to take full advantage of batteries, thereby smoothing the load on the bulk power grid.

The following are key lessons we learned during this work (see Table III for a synopsis).

- The performance results in Figure 11 demonstrate that our TMP can process and match trades much faster than what would typically be required in practice (see Section VI-E). This performance stems largely from the communication architecture provided by combining Ethereum and our RIAPS middleware.
- A key design challenge was resolving the conflict between safety, privacy, and efficiency. For example, the enforcement of feeder-level safety constraints required prosumers to reveal their feeder during trading, instead of being completely anonymous. Likewise, feeder-level safety constraints confounded meeting energy demand with local supply, even if there was surplus production in the microgrid (Figure 6).
- Feeder-level operational safety and stability constraints on trading were enforced via a blockchain-based smart contract (see Section V-B), in addition to the prosumer-level constraints enforced by tracking energy assets (see Section VI-C and [4]).
- Market efficiency and security were ensured by enabling the Ethereum smart contract to validate and evaluate the trading solutions that it receives (see Section V-B).
- Prosumer privacy was enabled by allowing them to hide their identity using anonymous addresses (see Section VI-C and [4]). To enable enforcing feeder-level safety constraints, however, prosumers still needed to reveal the feeder to which they belonged.

TABLE III
Requirements and Proposed Solutions

Requirement	Solution Approach
Communication Fabric	RIAPS [36] and Ethereum
Operational Safety and Stability	Feeder-level constraints modeled in the energy trading problem and enforced by the smart contract (Section V-B); prosumer-level constraints enforced by energy asset tracking (Section VI-C, [4])
Market Efficiency and Security	Objective modeled in the energy trading problem and enforced by the smart contract (Section V-B)
Privacy	Anonymous addresses (Section VI-C, [4])

Our future work will extend the approach presented in this paper to enable the energy trading system to recovery from failures by using a fault tolerance matching algorithm that ensures supply and demand can still be matched even if a certain number of physical elements fail or prosumers are disconnected. The datasets presented in this article are not readily available because as they are proprietary. This limits replicability, however, researchers are welcome to try the code with synthetic datasets available on our repository. Requests to access the datasets should be directed to the authors.

IX. Acknowledgements

We thank Prof. Gabor Karsai from Vanderbilt University and Prof. Srdjan Lukic for their feedback and help with the RIAPS platform. This manuscript has been released as a preprint [40] and also significantly extends content from a prior conference publication [41]. This work was funded in part by a grant from Siemens, CT and in part by grants from NSF under award number CNS-1647015, CNS-1818901, and CNS-1840052. The views presented in this paper are those of the authors and do not reflect the opinion or endorsement of Siemens, CT, or NSF.

References

[1] T. Randall, “The way humans get electricity is about to change forever,” Available at: https://www.bloomberg.com/news/articles/2017-09-13/the-way-we-get-electricity-is-about-to-change-forever, June 2015.

[2] F. Barbieri, S. Rajakaruna, and A. Ghosh, “Very short-term photovoltaic power forecasting with cloud modeling: A review,” *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 242 – 263, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136403211630733X

[3] O. Abrishambaf, F. Lezama, P. Faria, and Z. Vale, “Towards transactive energy systems: An analysis on current trends,” *Energy Strategy Reviews*, vol. 26, p. 100418, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2211467X19301105

[4] A. Laszka, A. Dubey, M. Walker, and D. Schmidt, “Providing privacy, safety and security in IoT-based transactive energy systems using distributed ledgers,” in *7th International Conference on the Internet of Things (IoT)*, October 2017.

[5] C. Cachin and M. Vukolić, “Blockchain Consensus Protocols in the Wild,” *arXiv:1707.01873 [cs]*, 2017. [Online]. Available: http://arxiv.org/abs/1707.01873

[6] J. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: Analysis and applications,” in *Advances in Cryptology - EUROCRYPT 2015*, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 281–310.

[7] P. Tholoniat and V. Gramoli, “Formal verification of blockchain byzantine fault tolerance,” *arXiv*, pp. arXiv–1909, 2019.

[8] K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, “Sawtooth: An introduction,” *The Linux Foundation*, 2018.

[9] Hyperledger, “The Fabric Model,” http://hyperledger-fabric.readthedocs.io/en/latest/fabric_model.html, 2017, accessed: April 24th, 2017.

[10] L. Orsini, Y. Wei, and J. Lubin, “Use of blockchain based distributed consensus control,” Apr. 13 2017, US Patent App. 15/292,783. [Online]. Available: https://www.google.com/patents/US20170103468

[11] “Power ledger whitepaper,” July 2017. [Online]. Available: https://powerledger.io/whitepaper/

[12] “Special Report: Block-chain world,” October 2017. [Online]. Available: https://spectrum.ieee.org/static/special-report-blockchain-world

[13] O. Dag and B. Mirafzal, “On stability of islanded low-inertia microgrids,” in *2016 Clemson University Power Systems Conference (PSC)*, March 2016, pp. 1–7.

[14] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, “”Designing microgrid energy markets: A case study: The Brooklyn Microgrid”,” *Applied Energy*, vol. 210, pp. 870 – 880, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030626191730805X

[15] Y. Amoussou-Guenou, A. Del Pozzo, M. Potop-Butucaru, and S. Tucci-Piergiovanni, “Dissecting tendermint,” in *Networked Systems*, M. F. Atig and A. A. Schwarzmann, Eds. Cham: Springer International Publishing, 2019, pp. 166–182.

[16] Tendermint, “What is Tendermint?” [Online]. Available: https://docs.tendermint.com/master/introduction/what-is-tendermint.html

[17] N. Szabo, “Formalizing and securing relationships on public networks,” *First Monday*, vol. 2, no. 9, Sep. 1997. [Online]. Available: https://journals.uic.edu/ojs/index.php/fm/article/view/548

[18] J. Bonneau, “What Bitcoin’s White Paper Got Right, Wrong and What We Still Don’t Know,” 2018, available at: https://www.coindesk.com/what-bitcoins-white-paper-got-right-wrong-and-what-we-still-dont-know.

[19] L. Newman, “$280M Worth of Ethereum Is Trapped Thanks to a Dumb Bug,” Available at: https://www.wired.com/story/280m-worth-of-ethereum-is-trapped-for-a-pretty-dumb-reason/, November 2017.

[20] K. Finley, “A $50 Million Hack Just Showed That the DAO Was All Too Human,” Available at: https://www.wired.com/2016/06/50-million-hack-just-showed-dao-human/, June 2016.

[21] G. B. Vieira, “A Blockchain Energy Trading Platform For Microgrids,” Master’s thesis, University of Southampton, 2018, available at: https://github.com/guibvieira/Microgrid-Blockchain-Project/blob/master/msc_dissertation_final.pdf.

[22] S. Nakamoto. (2008) Bitcoin: a peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[23] V. Buterin *et al.*, “Ethereum white paper,” 2013.

[24] N. Z. Aitzhan and D. Svetinovic, “Security and Privacy in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams,” *IEEE Transactions on Dependable and Secure Computing*, 2016.

[25] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” in *37th IEEE Symposium on Security and Privacy (S&P)*. IEEE, May 2016, pp. 839–858.

[26] P. McDaniel and S. McLaughlin, “Security and privacy challenges in the smart grid,” *IEEE Security & Privacy*, vol. 7, no. 3, 2009.

[27] C. Efthymiou and G. Kalogridis, “Smart grid privacy via anonymization of smart metering data,” in *1st IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2010, pp. 238–243.

[28] O. Tan, D. Gunduz, and H. V. Poor, “Increasing smart meter privacy through energy harvesting and storage devices,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1331–1341, 2013.

[29] A. Hussain, V. H. Bui, and H. M. Kim, “A Resilient and Privacy-Preserving Energy Management Strategy for Networked Microgrids,” *IEEE Transactions on Smart Grid*, 2017.

[30] B. P. Majumder, M. N. Faqiry, S. Das, and A. Pahwa, “An efficient iterative double auction for energy trading in microgrids,” in *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*. IEEE, December 2014, pp. 1–7.

[31] European Power Exchange, “EPEX SPOT operational rules,” 2017. [Online]. Available: http://www.epexspot.com/en/extras/download-center/technical_documentation

[32] D. Ilic, P. G. D. Silva, S. Karnouskos, and M. Griesemer, “An energy market for trading electricity in smart grid neighbourhoods,” in *6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, June 2012, pp. 1–6.

[33] M. N. Faqiry and S. Das, “Transactive energy auction with hidden user information in microgrid,” *arXiv preprint arXiv:1608.03649*, 2016. [Online]. Available: https://arxiv.org/abs/1608.03649

[34] S. Frank and S. Rebennack, “An introduction to optimal power flow: Theory, formulation, and examples,” *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016.

[35] IBM ILOG CPLEX, “V12. 1: User’s manual for CPLEX,” *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[36] https://riaps.isis.vanderbilt.edu/redmine/projects/riaps.

[37] P. Volgyesi, A. Dubey, T. Krentz, I. Madari, M. Metelko, and G. Karsai, “Time synchronization services for low-cost fog computing applications,” in *28th International Symposium on Rapid System Prototyping (RSP)*. IEEE, October 2017.

[38] S. Lab, “transactive-blockchain,” 2019. [Online]. Available: https://github.com/scope-lab-vu/transactive-blockchain

[39] M. A. Walker, A. Dubey, A. Laszka, and D. C. Schmidt, “PlaTIBART: a platform for transactive IoT blockchain applications with repeatable testing,” in *4th Workshop on Middleware and Applications for the IoT (M4IoT)*, December 2017.

[40] A. Laszka, S. Eisele, M. Walker, and K. Kvaternik, “Design and implementation of safe and private forward-trading platform for iot-based transactive microgrids,” 2017. [Online]. Available: https://pdfs.semanticscholar.org/402e/580d90132db773c69b74806cdc6bf1716683.pdf

[41] A. Laszka, S. Eisele, A. Dubey, G. Karsai, and K. Kvaternik, “Transax: A blockchain-based decentralized forward-trading energy exchanged for transactive microgrids,” *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 918–927, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/86450011