Privacy-Preserving Platform for Transactive Energy Systems

Karla Kvaternik
Siemens Corporate Technology
karla.kvaternik@siemens.com

Aron Laszka
Vanderbilt University
aron.laszka@vanderbilt.edu

Michael Walker
Vanderbilt University
michael.a.walker.1@vanderbilt.edu

Douglas Schmidt
Vanderbilt University
d.schmidt@vanderbilt.edu

Monika Sturm
Siemens Corporate Technology
monika.sturm@siemens.com

Martin Lehofer
Siemens Corporate Technology
martin.lehofer@siemens.com

Abhishek Dubey
Vanderbilt University
abhishek.dubey@vanderbilt.edu

Abstract

Transactive energy systems (TES) are emerging as a transformative solution for the problems faced by distribution system operators due to an increase in the use of distributed energy resources and a rapid acceleration in renewable energy generation. These, on one hand, pose a decentralized power system controls problem, requiring strategic microgrid control to maintain stability for the community and for the utility. On the other hand, they require robust financial markets operating on distributed software platforms that preserve privacy. In this paper, we describe the implementation of a novel, blockchain-based transactive energy system. We outline the key requirements and motivation of this platform, describe the lessons learned, and provide a description of key architectural components of this system.

Keywords Transactive energy platforms, blockchain, privacy, security, safety, smart contracts

ACM Reference format:
Karla Kvaternik, Aron Laszka, Michael Walker, Douglas Schmidt, Monika Sturm, Martin Lehofer, and Abhishek Dubey. 2017. Privacy-Preserving Platform for Transactive Energy Systems. In *Proceedings of ACM/IFIP/USENIX Middleware conference, Las Vegas, Nevada USA, December 2017 (Middleware'17),* 6 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 Introduction

Emerging Trends: Transactive energy systems (TES) have emerged as an anticipated outcome of the shift in electricity industry, away from centralized, monolithic business models characterized by bulk generation and one-way delivery, toward a decentralized model in which end users play a more active role in both production and consumption [10] [24]. In this paper, we consider a class of TES that operates in grid-connected mode. The main actors are the consumers, which are comprised primarily of residential loads and prosumers who operate distributed energy resources (DERs), such as rooftop, solar batteries or flexible loads capable of demand/response. Additionally, a distribution system operator (DSO) manages the grid connection of the network. Such installations are equipped with an advanced metering infrastructure consisting of TE-enabled smart meters. In addition to the standard functionalities of smart meters: i.e. the ability to measure line voltages, power consumption and production, and communicate these to the distribution system operator (DSO); TE-enabled smart meters are capable of communicating with other smart meters, have substantial on-board computational resources, and are capable of accessing the Internet and cloud computing services as needed. Examples of such installations include the well-known Brooklyn Microgrid Project, [3] and the Sterling Ranch learning community (currently under development) [12]. A key component of TES is a transaction management platform (TMP), which handles all market clearing functions in a way that balances supply and demand in the local market.

Why Blockchains?: The capabilities of TE-enabled meters allow them to form a blockchain (BC) based TMP executing a market mechanism, using smart contracts [29]. Examples of BC systems capable of executing smart contracts include Ethereum [7] and Hyperledger Fabric [9]. There are a number of appealing properties of BC systems that motivate their use in a TMP. Firstly, BC technology enables the digital representation of energy and financial assets, and their secure transfer from one set of parties to another. By design, the security of this value transfer is guaranteed by the interaction protocol itself and obviates the need for trusted transaction intermediaries. Secondly, the execution of smart contracts (i.e. code that captures the market logic and participant roles) is automated and guaranteed. Thirdly, the blockchain constitutes an immutable, complete, and fully auditable record of all transactions that have ever occurred in the BC system. These properties ensure market transparency, as well as the availability of a detailed market load profile, and grid utilization data. Thus, [1, 4, 27] have already considered such implementations.

Open challenges: Existing initiatives such as [1, 4, 27] do not consider the impact of the electricity market on the controller responsible for the stability of the system due to the expectation that the bulk grid will maintain system stability. Furthermore, although these solutions present interesting case studies, they provide only a subset of services, which do not affect the overall power flow on the grid in a significant way. For example, they do not address the security, stability, and privacy requirements, which we describe in the next section. The information technology backbone that allows energy trades in an open P2P market to take place anonymously, and securely, has yet to be developed [23, 28].

Middleware'17, Las Vegas, Nevada USA

DOI: 10.1145/nnnnnnn.nnnnnnn

Contributions → Design and Implementation of *Privacy-preserving Energy Transactions* (PETra) We focus on the specification of the components of the platform, their interfaces, and the distributed ledger in the system. Additionally, we highlight the architectural and protocol specifications of our platform, which ensure *privacy* and *security* for participants in the TES, as well as *safety* for the TES. The specific contributions of this paper are (a) proposal of a set of TES requirements extracted from our experience, (b) architectural and protocol specification and implementation of a new blockchain-based middleware called PETra, first introduced in [20], and (c) an empirical demonstration of the PETra functionality using an actual load profile data set from a microgrid test installation in Germany. In particular, the high volumes of microtransactions in the envisioned TES pose challenges related to real-time communication of sensor data; for example, request-reply messaging between TMP modules, and other signaling that occurs outside of the blockchain. We will refer to these as "out of band communications."

The outline of this paper is as follows. Section 2 describes the requirements for this class of distributed systems. Section 3 describes the state of the art. We present our solution in section 4. It is followed by an evaluation and discussion in Sections 4.3 and 4.4.

2 Requirement Analysis

The trading scenarios we consider involve consumers and prosumers that participate in a local P2P energy trading market by posting offers to sell produced energy, or offers to buy and consume energy in each consecutive time interval. An offer consists of quantity of energy being bought or sold, the time interval in which the trade is to be made, and possibly a reservation price - the maximum (or respectively, minimum) price at which the buyer (or respectively, seller) is willing to trade. The DSO bargains on the bulk market and provides all residual supply and demand within the microgrid.

We assume that each participant has a means of predicting her future power production and consumption based on historical data, and does so prior to trading on the market. An example of a home energy management system that provides this means is the Siemens Energy IP Analytics Suite. Moreover, each participant is represented by an automated trading agent that strategically posts offers to the TMP based on these predictions and the participant's personal trading goals.

In the simplest trading scenario, the DSO sets the price p per kWH for the local market; p is the price paid by any buyer and received by any seller, including the DSO. The DSO can then dynamically adjust the price p to affect the market efficiency, evaluated as the number of local transactions vs. energy demand being met from a bulk supplier. Another scenario includes a fully dynamic market where all sellers, including the DSO, post offers that include a reservation price. Each consumer then picks a selling offer on a first-come, first-served basis. An extension of this scenario involves double auctions where both selling and buying offers are posted to the TMP, which executes an automated, regulator-approved market clearing algorithm as an immutable smart contract on the TMP's blockchain system. This algorithm selects the clearing price p within each time interval. With respect to these trading scenarios we propose the following requirements.

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 the market algorithm must communicate clearing prices, buyer-seller matchings, or other market-related signals depending on the trading scenario. In order to meet the operational and safety requirements described next, these messages must be reliably delivered 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 P2P trading scenarios. Finally, the communication fabric must support confidentiality, integrity, and non-repudiation of transactional data.

Operational Safety and Cyber-Physical Security The trading activity permitted by the TMP shall not compromise the stability of the physical system operation. Moreover, congestion constraints along any feeder shall be respected.[1] This also requires assurance that malicious or negligent trading activity is discouraged. [2] Finally, the TMP should have provisions for preventing or detecting negligent or malicious interference with smart meters - i.e. the adversarial or natural attacks against the interface between the physical world and the blockchain; data logged shall be securely communicated to the DSO and requests made by the meter on behalf of the prosumer shall be accurately recorded on the blockchain.

Market Security The TMP shall include provisions for ensuring the protection of consumer interests, as well as those of the DSO. Consumer interests include being billed correctly and fairly based on energy prices set by the DSO and the measurements made by the smart meters. Additionally, it is important to ensure all prosumers will be allowed to participate in the market fairly.

Privacy Information such as the amount of energy produced, consumed, bought, or sold by any prosumer should be available only to the DSO and the essential market functions of the TMP. All bids and asks, and the matching thereof, should remain anonymous. A participant's energy usage patterns and personal information, such as financial standing, shall not be inferable from the participant's trading activity[3].

3 Analysis of State of the Art

The TMP system requires peer-to-peer messaging, enabling each stakeholder to receive all the required 'bid' messages, concerning a specific ask. Thereafter, a consumer can choose to accept a bid and inform the ledger about the acceptance. Once the bid is accepted, the transaction is recorded into a distributed ledger in a way that allows everyone in the community to agree that the transaction took place. Once consensus is established, the transaction is deemed successful, and we say that the market has cleared. In the context of this workflow, we next describe the state of the art across the two dimensions of Application and Communication platforms in smart grid and distributed transaction management platform for smart grid.

[1] In the context of grid-connected microgrids, system stability refers to real-time balancing – i.e. the system's ability to dynamically match supply and demand as closely as possible, and a tendency to drive the difference between supply and demand to zero under small perturbations. Resiliency refers to the system's ability to react to contingencies and recover from faults. Congestion on a transmission line occurs when the power flow exceeds the line's maximum rated capacity.

[2] Negligent trading may include producers who commit to a certain production level and fail to deliver. Transactional security means that the execution of contractual obligations among all participants, including the DSO, is guaranteed.

[3] Inference of energy usage patterns can be exploited by inferring the presence or absence of a person in their home, for example.

Application Platforms for Smart Grid There seem to be two approaches in general for moving power applications from centralized to distributed processing paradigms. One approach is to consider each remote computing entity (or node) as an *agent* [33] or *actor* [5] [21] that communicates via messages with other agents or actors, and focuses on specific grid issues such as state estimation, remedial action schemes, and load shedding. The other approach utilizes each remote computing entity as an open application platform that can host multiple applications managing varied aspects of the local grid [2]. Both approaches utilize messaging between nodes, and leverage a common set of services on each node, to handle distributed coordination concerns. [21] calls for group membership, leader election, voting, group consensus or agreement on data values, mutual exclusion on access to shared resources, and multicast communication with same order and atomic properties. Both [33] and [21] prototype their approaches using MATLAB toolkits, with [21] utilizing the Akka Java toolkit to model actors. [25] developed simulations using the SimPowerSystem software in the Simulink environment. Our application platform, called Resilient Information Architecture Platform for Smart Grid (RIAPS) [16], provides actor and component based abstraction, as well as support for deploying algorithms on devices across the network[4] and solves problems collaboratively by providing micro-second level time synchronization [32], failure based reconfiguration [13], and group creation and coordination services (still under active development), in addition to the services described in [21]. It is capable of handling different communication and running implemented algorithms in real-time.

Transaction Management Platforms (TMP) for Smart grid TMP require communication, as well as trading mechanisms that provide the capability to match the bids and asks. Additionally, they must provide fairness and integrity assurances. Blockchain based solutions have the potential to enable large-scale energy trading based on distributed consensus systems. However, popular blockchain solutions, such as Bitcoin [26] and Ethereum [8] suffer from design limitations that prevent their direct application to validating energy trades. In particular, their transaction-confirmation time is relatively slow and variable, primarily due to the proof-of-work algorithm and most of the communication occurring via the ledger. For example, Aitzhan and Svetinovic implemented a proof-of-concept platform for decentralized smart grid energy trading using blockchains, but their system is based on proof-of-work consensus, and they do not consider grid control and stability, or scalability [6]. Additionally, there is the problem of privacy - all transactions in these systems are public [19].

Most works in this area have focused on the privacy issue from the context of smart meters. McDaniel and McLaughlin discuss the privacy concerns of energy usage profiling, which smart grids could potentially enable [22]. Efthymiou and Kalogridis describe a method for securely anonymizing frequent electrical metering data sent by a smart meter [14] by using a third party escrow mechanism. Tan et al. study privacy in a smart metering system from an information theoretic perspective in the presence of energy harvesting and storage units [30]. They 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. However, the transaction data provides more fine-grained information than the smart meter usage patterns [18].

[4]RIAPS uses ZeroMQ [17], and Cap'n Proto [31], to manage the communication layer.

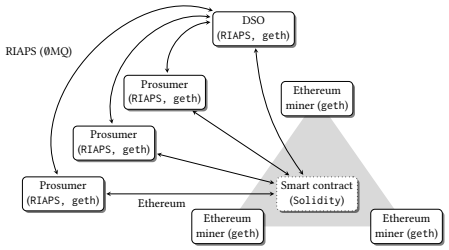

Figure 1. Components of PETra. The DSO and prosumers are comprised of RIAPS components and `geth` Ethereum clients. The smart contract is implemented in Solidity, a high-level language for Ethereum, and it is executed by a network of `geth` miners.

PETra extends these works by (1) leveraging a decentralized computation fabric provided by smart homes in the community, (2) addressing the privacy threat posed by trading using a novel trading sequence implementation, (3) showing how partial trades can be fulfilled, and (4) using off-blockchain communication primitives provided by the distributed application management platform RIAPS. While the conceptual design of PETra was presented in [20], this paper describes the revised protocol and the trading algorithm, and presents the implementation results.

4 Our Solution - PETra

Our system contains the following types of components (see Figure 1 for an illustration):

- DSO: There is a single component of this type, which represents the Distribution System Operator of the microgrid. The primary responsibilities of this component are ensuring the safe operation of the microgrid and regulating the total load of the microgrid. To this end, the DSO component can limit the energy and financial assets that the prosumers' withdraw for trading, and it can also set a price policy for the microgrid. Note that this component does not have to be online during trading, so the reliability of the system does not hinge on the reliability of this component.
- Prosumer: There is a component of this type for every household. The prosumer components are responsible for trading energy production and consumption for their households. To do so, a component first estimates the future production and consumption of the household, withdraws energy production or consumption assets from the DSO, and then trades these assets with other prosumers. To ensure that trading does not compromise the household's privacy, the component uses randomly generated anonymous addresses for trading, which hide the identities of trade partners from each other.
- Smart contract: This component (deployed as an Ethereum contract on the private blockchain) is responsible for keeping track of the energy and financial assets belonging to each anonymous address, enabling prosumers to post trade offers, and exchanging assets when another prosumer decides to take an offer. The contract is executed in a decentralized manner by a network of miners, which provides reliability. Additionally, we have several Ethereum clients, one per prosumer and one for the DSO, which interact with the smart contract.

4.1 Assets and Data Structures

The ability to specify points or intervals in time is crucial. For example, control signals specify how the microgrid load should change

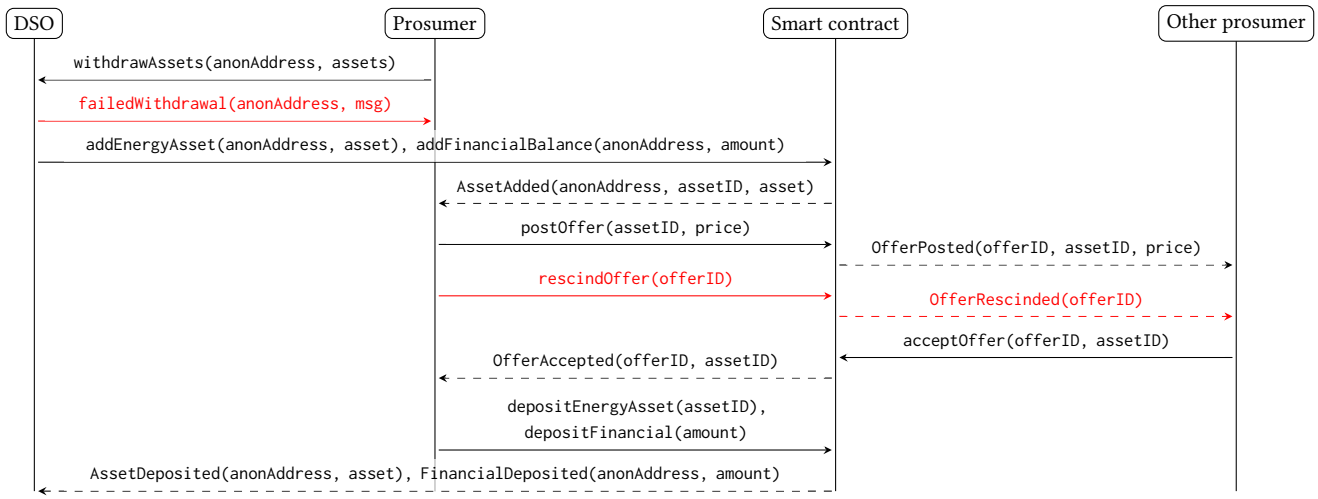

Figure 2. Sequence diagram of the trading workflow. Solid lines represent RIAPS messages and Ethereum transactions, while dashed lines represent smart-contract events. Messages and transaction in red stop the trading workflow.

at certain points in time, energy trades specify when energy will be consumed or produced, etc. To facilitate representing signals and transactions, we divide time into fixed-length intervals, and specify points or periods in time using these discrete timesteps. The length of the time interval is determined based on the timing assumptions of the physical power system. For example, the time interval may as low as 4 seconds, which corresponds to how frequently the control signal of the DSO typically changes [11].

Prosumers trade energy production and consumption with each other, which are represented in PETra by energy assets, which is a structure that comprises the following fields: (a) `int64 power`: non-negative amount of power to be produced or consumed (for example, measured in watts), (b) `uint64 start`: first time interval in which energy is to be produced (or consumed), and (c) `uint64 end`: last time interval in which energy is to be produced (or consumed). An asset with positive power value represents energy production, and we call it an `EnergyProductionAsset`. An asset with negative power value, on the other hand, represents energy consumption, and we call it an `EnergyConsumptionAsset`.

Energy trading must also involve the transfer of currencies, which are represented by financial assets. A `FinancialAsset` is simply an `uint64` value, denominated in a fiat currency.

4.2 Trading Workflow

Next, we discuss the trading workflow that is used by prosumers to trade energy production and consumption assets, as well as financial assets with each other. This workflow involves both off-blockchain messaging (using RIAPS), and on-blockchain transactions and events. We list these messages, transactions, and events in the order in which they typically appear in the workflow. Figure 2 shows a graphical illustration of the workflow.

- `withdrawAssets(anonAddress, assets)`: RIAPS 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 to protect her privacy. Before sending this message, the prosumer should generate a new random anonymous address. The message specifies the assets that the prosumer wishes to withdraw, and the anonymous address to which the DSO should transfer them, which must be cryptographically signed by the prosumer. Note that the prosumer may send this message long before actually engaging in trading, so the DSO does not have to be online continuously.
- `failedWithdrawal(anonAddress, msg)`: RIAPS message sent by the DSO to the prosumer, notifying the prosumer that the requested assets cannot be withdrawn due to, e.g., energy safety requirements or insufficient funds.
- `addEnergyAsset(anonAddress, asset)`, `addFinancialBalance(anonAddress, amount)`: smart contract transaction called by the DSO, creating energy and financial assets on the blockchain and transferring them to an anonymous address. Before recording this transaction, the DSO must first verify whether enabling the prosumer to trade these assets would violate any safety requirements. The transaction specifies the assets and the anonymous address to which they are transferred, and it must be cryptographically signed by the DSO.
- `AssetAdded(anonAddress, assetID, asset)`, `FinancialAdded(anonAddress, amount)`: events broadcast by the smart contract, notifying the prosumer that the requested assets have been transferred to the anonymous address.
- `postOffer(assetID, price)`: smart contract transaction called by a prosumer, publicly posting an energy bid or ask. If the prosumer is interested in buying energy, then it posts an energy bid, which specifies an energy consumption asset and a price. If the prosumer is interested in selling, then it posts an energy ask, which specifies an energy production asset and a price. vIn both cases, the transaction must be cryptographically signed by the private key of the address, and it locks the assets until the offer is accepted or rescinded.
- `OfferPosted(offerID, assetID, price)`: event broadcast by the smart contract, notifying prosumers that an offer was posted.
- `rescindOffer(offerID)`: smart contract transaction called by a prosumer, rescinding an offer. The transaction must be cryptographically signed by the private key of the poster.
- `acceptOffer(offerID, assetID)`: smart contract transaction called by a prosumer, accepting a previously posted offer. If the offer was an energy bid, then the other prosumer has to provide

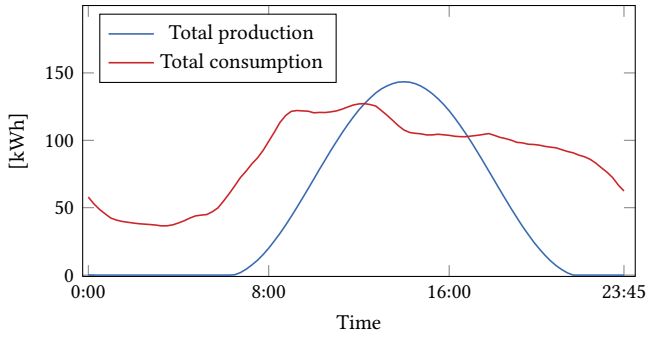

Figure 3. Load profile and Generation Profile in KWH per 15 minute interval. The horizontal axis shows time of day.

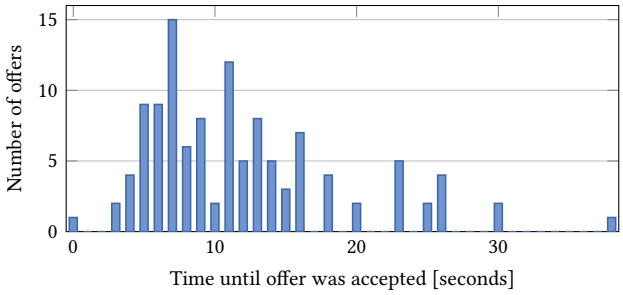

Figure 4. Histogram of time it takes to clear the two transactions related to post offer and accept offer. 90% of the trades were closed within 23 seconds or less.

an energy production assets; if the offer was an energy ask, then the other prosumer has to provide both energy consumption and financial assets. In both cases, the transaction must be cryptographically signed by the private key of the other prosumer's anonymous address. If there is an overlap between the time intervals of the offered asset, and the asset provided by the other prosumer, then the intersecting parts of the assets are exchanged and the non-overlapping parts are returned to their original owners. Similarly, based on the price and exchanged energy assets, a part of the financial asset is transferred to the seller, while the rest is returned to the seller.

- `OfferAccepted(offerID, assetID)`: event broadcast by the smart contract, notifying the prosumer that its offer has been accepted, and the assets have been exchanged.
- `depositEnergyAsset(assetID)`, `depositFinancial(amount)`: smart contract transactions called by a prosumer, depositing energy and financial assets to the prosumer's account. The transaction specifies the assets, and it must be cryptographically signed by the anonymous address that owns them. Note that to protect privacy, the transaction does not specify the prosumer, so the DSO has to keep track of which prosumer has used which anonymous address.
- `AssetDeposited(anonAddress, assetID)`, `FinancialDeposited(anonAddress, amount)`: event broadcast by the smart contract, notifying the DSO that assets have been deposited from anonymous address, which triggers the transfer of these assets to the prosumer's account at the DSO.

4.3 Case Study

We use data collected by Siemens, from a microgrid in Germany, to demonstrate a simulated transactive scenario. Figure 3 shows the total energy produced in this system over the day, and the total energy consumed. We use a $T = 15$ minute time interval for bids and asks. We picked a 3.5 hour time interval (from 2:15pm to 5:45pm) and 2 producers and 7 consumers that overlapped with the peak of production capacity (see Figure 3). We ran the network across six virtual machines, each with 2 virtual CPUs, 4 GB RAM and 40 GB hard-disk. The `geth` clients (one per actor) and miners were equally distributed on this network. The actors (Prosumers and DSO) were written in Python, and they communicated with the `geth` clients using JSON-RPC API provided by Ethereum. The actors communicated with each other using RIAPS and polled the blockchain ledger for transactional updates using custom filters, which are supported by the Ethereum API.

Figure 4 shows the distribution of the time between when an offer was made by a producer, and then the time when the offer was accepted by a consumer and cleared. As shown by Figure 2, this includes two transactions, `postOffer` and `acceptOffer`, which have to be verified and recorded by the miners. To clear these two transactions, at least two blocks need to be mined. The statistics of the clearing-time distribution are as follows: average = 11.79 seconds, median = 11 seconds, variance = 46.74, maximum = 38 seconds, minimum = 0 seconds, and 90% of trades were cleared within 23 seconds or less.

4.4 Requirements Analysis Discussion

We conclude this section with a brief discussion of how PETra satisfies the requirements outlined in Section 2.

Communication Fabric: The key requirements for communication are reliability and security. In PETra, communication and messaging services are built on (a) RIAPS between the DSO and prosumers, and (b) blockchain transactions and events between the smart contract and other components. The RIAPS communication layer [15] presents a reliable messaging service, which is being currently extended to provide message confidentiality, integrity, and non-repudiation with the help of digital signatures. Since communication between prosumers and the DSO (i.e. withdrawal) may happen well in advance of actual trading, the DSO and the messaging service do not have to be online continuously. Combined with the features of RIAPS, this flexibility in uptime leads to a very high level of reliability.

For messaging between the smart contract and other components, the blockchain provides a secure and reliable communication medium. The blockchain ledger is an immutable, complete, and fully auditable record, which guarantees integrity and non-repudiation for transactions and events. Note that—by design—the blockchain does not provide confidentiality, since every transaction and event is public; we will discuss privacy implications and requirements in detail below. Finally, the blockchain provides a high level of reliability since the ledger is maintained by multiple nodes, which can reach consensus even in the presence of some misbehaving or malicious nodes.

Operational Safety, Cyber-Physical Security, and Market Safety: For safety and cyber-physical security, it is crucial to ensure that trading activity cannot compromise the stability of the grid and congestion constraints are respected. PETra achieves these goals by enabling the DSO to tightly control the amount of energy that a prosumer may (offer to) sell or buy. A prosumer's energy trading

workflow (see Figure 2) always begins with a withdrawal from the DSO. By limiting the amount assets that can be withdrawn, the DSO limits the bids and asks that may be posted by a prosumer, thereby enforcing safety requirements (e.g., preventing a prosumer from offering to produce more power than her production capacity). Fine-grained withdrawal rules based on time, power, etc, can be used to prevent a wide range of negligent or malicious trading.

To protect the prosumers' interests, we must enable them to detect and prove if they are incorrectly billed or denied fair participation in the market. PETra meets these goals due to the public, fully auditable, and immutable nature of the blockchain ledger.

Privacy: Privacy requirements dictate that prosumers cannot gain information regarding other prosumers' consumption and production—not even if they are trade partners. This requirement presents an interesting challenge since every transaction on the blockchain ledger is public. PETra provides privacy through pseudonymous trading; instead of real identities, prosumers use randomly chosen addresses for trading with each other. However, pseudonymous addresses could be de-anonymized either by (a) learning which addresses belong to the same prosumer or (b) using the prosumers' communication addresses (e.g., IP addresses used to send transactions). Firstly, by employing a large number of anonymous addresses, a prosumer can effectively prevent de-anonymization attacks that would link her addresses together.[5] Secondly, by combining our platform with a communication anonymity solution, such as onion routing, we can prevent de-anonymization based on communication addresses.

5 Conclusions and Future Work

A transaction management platform (TMP) is the key component of a transactive energy system. The role of the TMP is to facilitate the deployment of applications that help maintain stability of the microgrid, as well as implement efficient market mechanisms and enable an open P2P energy trading market. In this paper we proposed a blockchain-based TMP called PETra, which extends existing works by (1) leveraging a decentralized computation fabric provided by the smart homes in the microgrid, (2) addressing the privacy threat posed by trading using a novel trading sequence implementation, (3) showing how partial trades can be fulfilled, and (4) using off-blockchain communication primitives provided by the distributed application management platform RIAPS.

During the experiments, we made the following observations: Blockchains are not enough by themselves to implement a full-fledged TMP for transactive energy systems. We need off-blockchain communication for (1) performance (transactions may be slow) (2) reliability (transactions may be lost before they are permanently recorded) (3) privacy (we can anonymize assets by mixing on the blockchain, but doing it off-blockchain is much more efficient). Additionally, existing smart contract languages (e.g., Solidity) have some serious limitations (and peculiarities), which complicate the implementation of some domain logic. For instance, at the time of writing, Solidity did not provide floating-point data types.

References

[1] 2002. TenneT unlocks distributed flexibility via blockchain. (May 2002). https://www.tennet.eu/news/detail/tennet-unlocks-distributed-flexibility-via-blockchain/

[2] 2014. FREEDM System Layered Architecture. Presentation. (April 2014).

[3] 2017. Brooklyn Microgrid. (2017). http://brooklynmicrogrid.com/

[4] 2017. Power Ledger Whitepaper. (Jul 2017). https://powerledger.io/whitepaper/

[5] Gul A Agha. 1985. *Actors: A model of concurrent computation in distributed systems.* Technical Report. MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB.

[6] Nurzhan Zhumabekuly Aitzhan and Davor Svetinovic. 2016. Security and Privacy in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams. *IEEE Transactions on Dependable and Secure Computing* (2016).

[7] Vitalik Buterin. 2013. Ethereum white paper - a next generation smart contract and decentralized application platform. (2013). http://bitpaper.info/paper/5634472569470976

[8] Vitalik Buterin and others. 2013. Ethereum white paper. (2013).

[9] Christian Cachin. Architecture of the Hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers* (2016). https://pdfs.semanticscholar.org/f852/c5f3fe649f8a17ded391df0796677a59927f.pdf

[10] E. Cazalet, P. De Marini, J. Price, E. Woychik, and J. Caldwell. 2016. *Transactive Energy Models.* Technical Report. National Institute of Standards Technology.

[11] Federal Energy Regulatory Commission and others. 2011. Frequency regulation compensation in the organized wholesale power markets. *Order* 755 (2011), 76.

[12] Sterling Ranch Development Company. 2017. The Nature of Sterling Ranch. (2017). http://sterlingranchcolorado.com/about/

[13] Abhishek Dubey, Gabor Karsai, and Subhav Pradhan. 2017. Resilience at the edge in cyber-physical systems. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on.* IEEE, 139–146.

[14] Costas Efthymiou and Georgios Kalogridis. 2010. Smart grid privacy via anonymization of smart metering data. In *1st IEEE International Conf. on Smart Grid Communications (SmartGridComm).* IEEE, 238–243.

[15] Scott Eisele, Istvan Madari, Abhishek Dubey, and Gabor Karsai. 2017. RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In *20th IEEE International Symposium on Real-Time Computing.* IEEE.

[16] S. Eisele, I. Mardari, A. Dubey, and G. Karsai. 2017. RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC).* 125–132. DOI: https://doi.org/10.1109/ISORC.2017.22

[17] Pieter Hintjens. 2010. ZeroMQ: The Guide. *URL http://zeromq. org* (2010).

[18] A. Hussain, V. H. Bui, and H. M. Kim. 2017. A Resilient and Privacy-Preserving Energy Management Strategy for Networked Microgrids. *IEEE Transactions on Smart Grid* PP, 99 (2017), 1–1. DOI: https://doi.org/10.1109/TSG.2016.2607422

[19] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *Security and Privacy (SP), 2016 IEEE Symposium on.* IEEE, 839–858.

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

[21] H. Lee, S. Niddodi, A. Srivastava, and D. Bakken. 2016. Decentralized voltage stability monitoring and control in the smart grid using distributed computing architecture. In *2016 IEEE Industry Applications Society Annual Meeting.* 1–9. DOI: https://doi.org/10.1109/IAS.2016.7731871

[22] Patrick McDaniel and Stephen McLaughlin. 2009. Security and privacy challenges in the smart grid. *IEEE Security & Privacy* 7, 3 (2009).

[23] R. Melton and J. Fuller. 2016. Transactive Energy: Envisioning the Future. *IEEE Electrification Magazine* 4, 4 (Dec 2016), 2–3. DOI: https://doi.org/10.1109/MELE.2016.2614198

[24] R. B. Melton. 2013. *Gridwise transactive energy framework.* Technical Report. Pacific Northwest National Laboratory.

[25] F. Meng, R. Akella, M. L. Crow, and B. McMillin. 2010. Distributed Grid Intelligence for future microgrid with renewable sources and storage. In *North American Power Symposium 2010.* 1–6. DOI: https://doi.org/10.1109/NAPS.2010.5618963

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

[27] L. Orsini, Y. Wei, and J. Lubin. 2017. Use of Blockchain Based Distributed Consensus Control. (April 13 2017). https://www.google.com/patents/US20170103468 US Patent App. 15/292,783.

[28] S. M. Sajjadi, P. Mandal, T. L. B. Tseng, and M. Velez-Reyes. 2016. Transactive energy market in distribution systems: A case study of energy trading between transactive nodes. In *2016 North American Power Symposium (NAPS).* 1–6. DOI: https://doi.org/10.1109/NAPS.2016.7747895

[29] N. Szabo. 1997. Formalizing and securing relationships on public networks. *First Monday* 2, 9 (1997).

[30] Onur Tan, Deniz Gunduz, and H Vincent Poor. 2013. Increasing smart meter privacy through energy harvesting and storage devices. *IEEE Journal on Selected Areas in Communications* 31, 7 (2013), 1331–1341.

[31] Kenton Varda. 2015. CapâĂŽn Proto. (2015).

[32] Peter Volgyesi, Abhishek Dubey, Timothy Krentz, Istvan Madari, Mary Metelko, and Gabor Karsai. 2017. Time Synchronization Services for Low-cost Fog Computing Applications. In *Rapid Systems Prototyping (RSP).* IEEE, IEEE.

[33] J. Xie, C.-C. Liu, and Sforna. M. 2015. Distributed Underfrequency Load Shedding Using a Multi-Agent System. In *IEEE PowerTech.*

[5] Note that generating new addresses is trivial.