

# A Blockchain-based Platform for Exchange of Solar Energy: Laboratory-scale Implementation

Manisa Pipattanasomporn, Murat Kuzlu and Saifur Rahman  
*Bradley Department of Electrical and Computer Engineering, Advanced Research Institute,  
Virginia Tech, USA*

**Abstract**— Over the last decade, the increasing uptake of rooftop solar photovoltaics (PV) at the grid edge transforms residential houses into complex energy “prosumers”. A house with rooftop solar PV can both consume and export electricity. Hence, it can participate in a “Transactive Energy” network involving peer-to-peer (P2P) exchange of excess electricity. The challenge is to keep track of these transactions and compensate buyers and sellers accordingly. Recently, blockchain has emerged as a distributed ledger technology which allows exchanges among participants without the need for a central market entity. There are several blockchain pilots in the energy sector, which focus on the business, legal and financial aspects, but without much details about how to implement a blockchain-based P2P trading platform. This paper addresses this issue by presenting laboratory-scale implementation of a blockchain network for exchange of solar electricity among participants using Hyperledger – an open-source collaborative effort. Participants, assets and transactions necessary to establish the blockchain-based network for keeping track of solar PV output exchanges are described, together with the smart contract, use cases and their implementation.

**Keywords**— P2P trading, solar PV, blockchain, hyperledger.

## I. INTRODUCTION

Traditionally, an electric power system relies on the centralized model of electricity supply where an electric utility is the central authority responsible for distributing electricity generated from large-scale power plants. Due to the continuing decline in photovoltaic (PV) module and system prices, a large number of rooftop solar PV units have been deployed over the past decade. As of 2017, there are more than one million American homes with rooftop solar PV units [1, 2]. Due to their ability to both produce and sell excess electricity, an individual house has now become a more complex energy “prosumer”.

Beginning in 2013, blockchain [3] has emerged as a distributed ledger technology that can track exchanges among participants without the need for a central clearinghouse. There are several research studies that consider blockchain-enabled peer-to-peer (P2P) energy trading as a viable alternative to traditional power delivery from a central electric utility. Authors in [4, 5] perform the review of P2P energy trading projects, several of which run on the blockchain technology. Examples are: PowerLedger, Brooklyn Microgrid, Dajie, Share&Charge, NRGcoin, SolarCoin, Thesunexchange, Electron, and Enerchain. The PowerLedger project [6] offers a P2P marketplace for renewable energy. Its whitepaper [7] provides high-level discussion of the system architecture and the token ecosystem. The Brooklyn microgrid project [8], developed by New York-based startup, LO3 Energy, aims at creating a blockchain-based P2P trading system allowing

electricity to be directed to hospitals, shelters and community centers when needed. The Dajie project [9] provides a blockchain-based platform that allows P2P energy exchange for a community, redemption of carbon credit for prosumers, and payment of energy and services for energy companies. The Share&Charge project [10] offers an open-network EV charging solution developed on the Ethereum blockchain [11] that allows seamless access to charging poles across different countries. The NRGcoin project [12] provides rewards for producing green energy (1kWh = 1NRGcoin). Its concept is similar to that of the SolarCoin project [13], where one SolarCoin represents one MWh of solar electricity generation (1MWh = 1SolarCoin). The Sun Exchange project [14] allows individuals to buy solar cells, lease them, mainly in Africa, and arrange monthly lease rental collection and distribution in Bitcoin or local currency. The Electron project [15], initiated in UK and running on Ethereum, offers a flexible blockchain-based trading platform supporting electricity, gas, as well as community energy projects. The Enerchain project [16] focuses on P2P trading in the wholesale energy market using the blockchain technology. There are also several other blockchain trials that have also been initiated in the energy sector, such as LINQ, Blockcharge and WePower. LINQ [17] is a blockchain service that allows solar power generators to sell PV generation certificates to anyone who wishes to subsidize solar energy. Blockcharge [18] offers blockchain-powered charging stations for electric cars across Germany. WePower [19] is a blockchain-based green energy trading platform that enables green energy producers to raise capital by issuing tradable energy tokens. In all of these projects, high-level documentations detailing business, legal and financial issues are available. However, details on technical implementation of a blockchain-based P2P trading platform are missing.

In terms of research study, in [20], authors perform a preliminary economic evaluation of the market mechanism of blockchain-based local energy markets. In [21], authors compare the P2P electricity trading in terms of business models and their characteristics, and emphasize that the P2P electricity trading can become more popular in the future due to changes in the new electricity trading environment. In [22], authors demonstrate a blockchain implementation in a campus environment for transactive energy auctions. Authors in [23] discuss components of microgrid energy markets related to the Brooklyn microgrid project. Authors in [24] discuss business models to ensure transparency of energy prosumers. Several other energy-related applications made possible with the blockchain technology include, for example, the use of blockchain in islanded microgrids to annotate energy losses [25] and support emission trading applications [26]. Authors in [27] also reviews how blockchain technology works in the Internet-of-Thing (IoT) setting.

It can be seen that almost all of the previous blockchain-related work primarily focuses on the market and business aspects and do not provide technical implementation details. Hence, it is the focus of this paper to provide such technical details describing how a blockchain-based energy trading platform for exchanging solar PV output is designed, developed and implemented using the open-source Hyperledger fabric blockchain framework [28]. Participants, assets, transactions and the smart contract of the P2P trading platform for exchange of solar energy are discussed in details. Additionally, the P2P trading implementation in a laboratory environment is elaborated that obtains real-time readings from a 6.4kW rooftop solar PV connected with a smart inverter. Overall, this paper serves as a guideline for researchers and utility engineers to develop, deploy and experiment on a blockchain network, and sets the stage for future development of blockchain-based P2P trading of solar energy.

## II. THE PROPOSED BLOCKCHAIN-BASED P2P ENERGY TRADING PLATFORM

The proposed blockchain-based P2P energy trading platform is designed to support the exchange of excess solar PV output in a neighborhood. The high-level architecture of a blockchain-based platform is depicted in Fig. 1, which illustrates the following major blockchain components.

- ‘Peer’ is owned and maintained by each group of participants. ‘Peer’ is a network entity that commits transactions and manages/maintains a ledger, on which a smart contract runs to encode assets and transaction instructions (i.e., business logic agreed by participants in the network).
- ‘Ordering Service (OS)’ is a node that provides a shared communication channels to participants, offering a broadcast service for messages containing transactions and implementing service delivery guarantees.
- ‘Certificate Authority (CA)’ provides registration of identities & issue enrollment certificates for participants

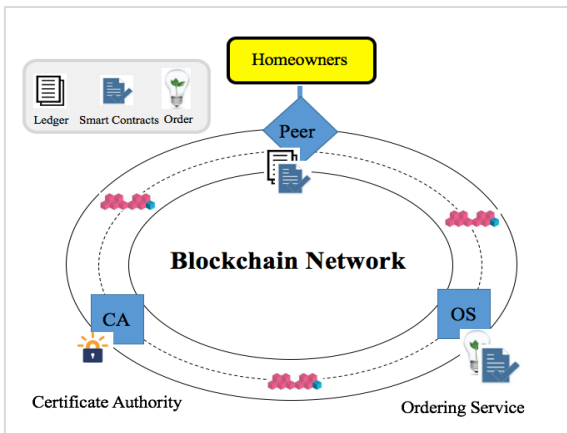


Fig. 1. High-level architecture of a blockchain-based platform.

In this study, participants, assets and transactions are defined in the ‘Peer’ of the proposed blockchain-based P2P trading platform. These are summarized in Table I, together with their corresponding parameters.

TABLE I.  
PARTICIPANTS, ASSETS AND TRANSACTIONS OF THE PROPOSED BLOCKCHAIN-BASED P2P TRADING PLATFORM

Participants	Assets	Transactions
Homeowner	PV	AcceptOfferBroadcast
➤ balance (token)	➤ PVID	➤ listingid
➤ email	➤ owner	PVOffer
➤ firstName	KWhlisting	➤ reservePrice
➤ lastName	➤ listingid	➤ kWhavailable
	➤ state	➤ listingid
	➤ PVOffer	➤ owner
	➤ BUYOffer	BUYOffer
		➤ BidPrice
		➤ kWhQuantity
		➤ listingid
		➤ homeowner
		CloseBidding
		➤ listingid

### A. Participants

In this context, participants are defined as those involved in the buying/selling of excess solar PV generation in a neighborhood, i.e., homeowners, which are of two types:

- prosumer – a homeowner with rooftop solar PV whose house can both consume and produce electricity; and
- consumer – a homeowner without PV whose house can only consume electricity.

Each homeowner is registered using the following parameters: ‘balance’, ‘email’, ‘firstName’ and ‘lastName’. The ‘balance’ specifies the homeowner’s available digital currency (i.e., tokens), which is used to buy/sell solar electricity in this blockchain network. The homeowner is identified by a unique ID, which is an email address. When a transaction (buy/sell electricity) occurs, the number of tokens is updated in participants’ digital wallets.

### B. Assets

In this blockchain network, two types of assets are defined: ‘PV’ and ‘kWhlisting’.

The ‘PV’ asset lists all PV registered by homeowners. All homeowners who would like to sell their excess PV generation must register their PVs as an asset in the blockchain network with a unique ‘PVID’ and the unique ‘owner’ ID, which is an email address.

The ‘kWhlisting’ asset is automatically generated by the system every hour, specifying ‘listingid’, ‘state’, ‘PVOffer’ and ‘BUYOffer’. The ‘listingid’ specifies date and time of PV electricity exchange. There are two options for ‘state’: ACCEPT\_OFFERS, indicating that offers from sellers/buyers are now accepted, and AUCTION\_CLOSED, indicating that the auction is now closed. ‘PVOffer’ and ‘BUYOffer’ are the offers from sellers/buyers (which are defined under ‘Transactions’, discussed next). In short, this asset gathers the offers to sell excess PV generation (kWh) from all prosumers in a neighborhood at a particular hour, as well as gathers the bids to buy energy from PV from consumers at the same hour.

### C. Transactions

In this work, four types of transactions are defined, which facilitate exchange of excess PV generation, including ‘AcceptOfferBroadcast’, ‘PVOffer’, ‘BUYOffer’ and

‘CloseBidding’. Each transaction requires the ‘listingid’ parameter, which specifies the date and time of solar electricity exchange.

- ‘AcceptOfferBroadcast’ – This is the broadcast message sent to all participants automatically by the system at the beginning of each hour. The broadcast message informs all homeowners that the system is now open to accept offers to buy/sell excess PV generation, i.e., the start of the bidding period.
- ‘PVOffer’ – After receiving the broadcast message, prosumers (identified by homeowners’ unique ID or email addresses) can send the offers to sell their excess PV generation (‘kWhavailable’, kWh) with an optional reserve price (‘reservePrice’, token/kWh).
- ‘BUYOffer’ – Once receiving the broadcast message, consumers (identified by homeowners’ unique ID or email addresses) can send the bids to buy energy from PV specifying the amount of energy needed (‘kWhQuantity’, kWh) and the price they are willing to pay (‘BidPrice’, token/kWh).
- ‘CloseBidding’ – This transaction specifies that the bidding period is now closed, and the designed smart contract is now initiated to process the PV trading among homeowners.

The timeline of each transaction is illustrated in Fig. 2. Assuming that the intended exchange of excess PV output is between 13:00-14:00 on Mar 21, the ‘AcceptOfferBroadcast’ message is sent out to all participants at 12:01, indicating that the system now accepts buy/sell offers. Anytime between 12:01 and 12:59, homeowners can submit the offers to sell their excess generation or the bids to buy PV generation for the next hour 13:00-14:00. The ‘CloseBidding’ transaction is initiated at 12:59 to clear the market. The actual exchange of PV generation occurs between 13:00-14:00, together with the recording of actual kWh sales and kWh consumption.

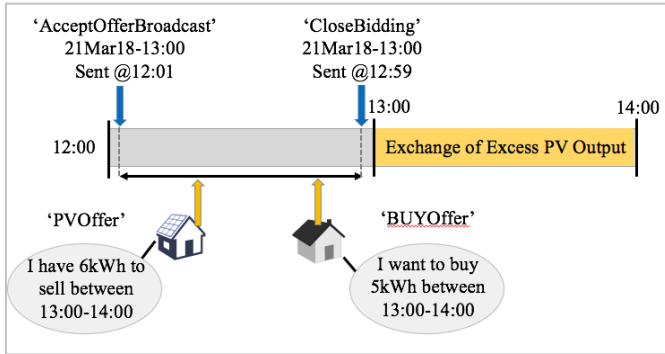


Fig. 2. Transaction timeline examples.

#### D. Smart Contract

In the proposed P2P trading platform, the smart contract defines how the market is cleared taking into account multiple offers to buy/sell PV generation. The smart contract code is embedded in the ‘CloseBidding’ transaction, and is called when the ‘CloseBidding’ transaction is initiated. In this study, a simple smart contract is implemented, and its flowchart is illustrated in Fig. 3.

Once the ‘CloseBidding’ transaction is called, offers from prosumers and consumers are sorted. PVOffers are sorted according to their offer time recorded in the system. BUYOffers

are sorted from high to low according to the multiplication of the bid price (token/kWh) and the quantity to purchase (kWh).

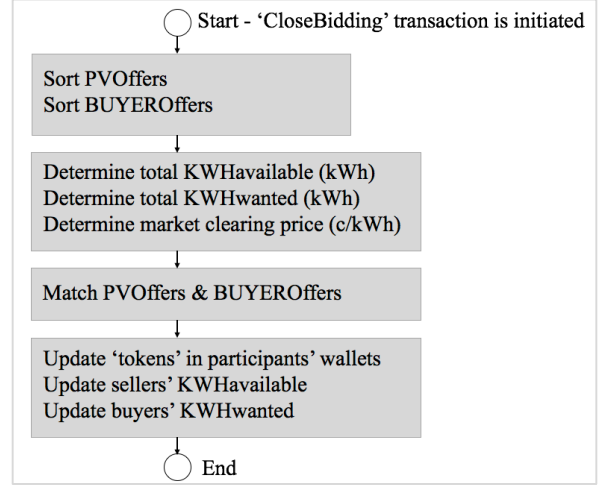


Fig. 3. Smart contract flowchart.

Based on PVOffers and BUYOffers, the total kWh available for sales (i.e., kWhavailable) and kWh to be purchased (i.e., kWhwanted) are calculated. In this experiment, the market clearing price (MCP, token/kWh) at each hour is simply determined using the averaged bid price offered by all buyers, as shown in (1). Note that: any method to determine MCP can be used. Since this study focuses on describing the technical implementation of a P2P energy trading, a simplified method of calculating MCP is used.

$$MCP_t = \frac{\sum_{i=1}^{N_t} (\beta_{n,t} \times kWh_{n,t})}{\sum_{i=1}^{N_t} (kWh_{n,t})} \quad (1)$$

Where,

- $MCP_t$  : Market clearing price at hour  $t$  (token/kWh)
- $\beta_{n,t}$  : Bid price offered by homeowner  $n$  at hour  $t$  (token/kWh)
- $kWh_{n,t}$  : kWh to be purchased by homeowner  $n$  at hour  $t$
- $N_t$  : Total number of homeowners who want to buy PV output

To match PVOffers and BUYOffers, this study considers that an offer to sell excess PV generation that is received first will be sold first, i.e., first-in, first-out (FIFO); and that the highest offer to buy PV generation (i.e., bid price \* quantity) will be met first. The market is cleared using the MCP indicated in (1). Tokens are now updated for all participants, so are the sellers’ amount of kWh sales and the buyers’ amount of kWh purchased.

### III. IMPLEMENTATION OF A BLOCKCHAIN NETWORK USING THE OPEN-SOURCE HYPERLEDGER FRAMEWORK

The proposed P2P trading platform was implemented using the open-source Hyperledger Fabric blockchain framework [28]. It includes the following three steps: (1) set up the Hyperledger Composer environment; (2) set up the REST API server with multiple users; and (3) create identities and wallets for participants.

### A. Set up the Hyperledger Composer Environment

In this study, the Hyperledger Composer environment was selected for conducting this experiment. Hyperledger Composer [29] is a development framework built upon the open-source Hyperledger Fabric blockchain infrastructure. It can be used to easily create and test blockchain-based business networks. A computer running Ubuntu 16.04 LTS operating system was used. Installation of the Hyperledger Composer pre-requisites [30] was carried out, including Docker Engine, Docker-Compose, Node, npm, git and Python: 2.7.x, and the Hyperledger Composer development environment was then installed [31]. Several command line interface (CLI) tools were also installed, including ‘composer-cli’ (contains all essential operations), ‘generator-hyperledger-composer’ (a utility for generating application assets), ‘Yeoman’ (a tool for generating applications) and ‘composer-rest-server’ (a utility for running a REST server to expose the business networks as RESTful APIs). Then, ‘composer-playground’ was installed as a browser app for editing, testing and deploying business networks.

After installing the development environment and running the Hyperledger Composer Playground, the business network named ‘pvtrading’ with its definition was created. In this experiment, the blockchain instance of a Hyperledger Composer was set up with a specific IP address at port 8080 (IP: xx.xx.xx.xxx:8080). The ‘pvtrading’ business network composes of the following files, defining the participants, assets, transactions and the smart contract, previously specified in Section II:

- The model file (‘*model.cto*’) defines the definition of assets, participants and transactions using Hyperledger Composer modeling language (as indicated in Table I).
- The script file (‘*logic.js*’) contains the logic of transaction functions and definition of the smart contract (as indicated in Fig. 3).
- The access control file (‘*permissions.acl*’) defines permissions and rules for each user to perform selected actions in the business network.

### B. Set up REST API server with multiple users

After setting up the Hyperledger Composer environment, the Hyperledger Composer REST (REpresentational State Transfer) server was deployed to expose the ‘pvtrading’ business network as RESTful application program interface (APIs). In this experiment, an instance of a Hyperledger Composer REST server was set up on the same machine as the Composer Playground at port 3000 (IP: xx.xx.xx.xxx:3000/explorer). The REST server provides RESTful API, which uses HTTP requests (e.g., GET, PUT, POST and DELETE) to allow participants to interact with the deployed blockchain-based P2P platform.

In this experiment, the REST server was configured in the multiple user mode to distinguish each client of the REST server. The multiple user mode provides Blockchain ‘identities’ (i.e., business network cards) and ‘wallet’ (to hold business network cards of authenticated users) to all participants in the blockchain network for digitally signing transactions. This is discussed in more details in the next subsection (Section III(C)).

In the multiple user mode, the OAuth authorization was implemented via Github to authorize each client’s connection to the REST server with a specific clientID and clientSecret. Doing so, an access token is displayed. This access token is used to authenticate each participant (i.e., REST client) to the REST server by specifying this access token in the query string (?access\_token=xxxxxx). With OAuth authorization, the deployed blockchain-based P2P platform can now securely support multiple buyers and sellers in a secure fashion.

### C. Create Identities and Wallets

In the Hyperledger Composer each participant has an identity. A blockchain identity is in the form of identity documents, i.e., business network card (filename: \*.card), which are used to cryptographically sign the transactions submitted to the business network. In this study, such identities were issued manually and sent to the participants. Before being able to interact with the business network, e.g., submitting PVOffer and BUYOffer, participants need to add their identities to their wallets.

## IV. P2P TRADING IN A LABORATORY ENVIRONMENT

To demonstrate the proposed P2P trading concept, an experiment was set up in a laboratory environment at the Virginia Tech - Advanced Research Institute in Arlington, Virginia, USA.

### A. Laboratory Set up

The laboratory set up is depicted in Fig. 4.

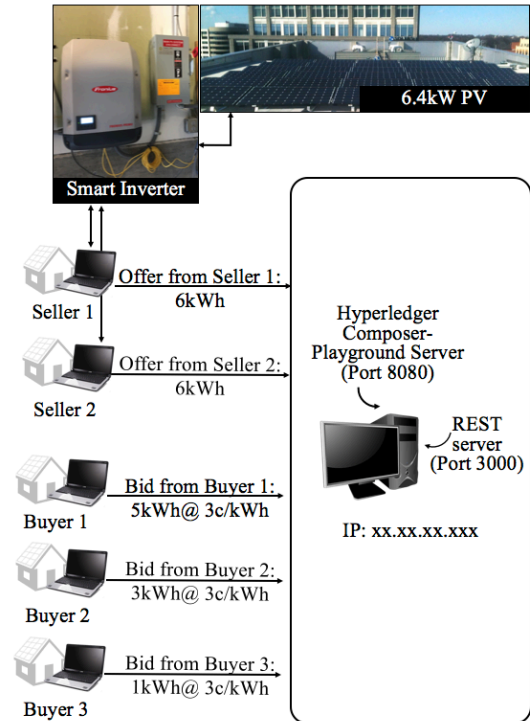


Fig. 4. Laboratory setup.

In this set up, one computer running the Linux-Ubuntu operating system was used to host the Hyperledger Composer-Playground server and the Hyperledger Composer REST server. The Hyperledger Composer Playground and the REST server were



set up according to the discussion in Section III. A dedicated IP address (IP: xx.xx.xx.xxx) was provided to this computer so that it could serve as the dedicated server hosting the deployed blockchain-based P2P trading platform. The Hyperledger Composer-Playground server was running on this computer at port 8080, and the REST server at port 3000.

Five computers were used to represent five participants, i.e., two prosumers (sellers) and three consumers (buyers). Each participant was provided a unique business network card. Once these business network cards were imported into participants' wallets, these sellers and buyers could then interact with the deployed P2P trading platform.

An instance of a Python code was run on each of these computers (each computer has a unique IP address) to perform selected actions on behalf of each seller/buyer. Selected actions are, for example, listening to the broadcast message (i.e., 'AcceptOfferBroadcast') from the Hyperledger-Composer Playground server, submitting an offer/bid and forecasting solar PV output. This topic is further elaborated in Section IV(B), describing P2P trading use cases.

The 6.4kW solar PV with a smart inverter located on the rooftop of this building was also a part of this experiment. Sellers were set up to retrieve real-time PV output data from the 6.4kW rooftop PV unit through an Ethernet-based, open JSON interface integrated in the smart inverter. Raw data retrieved from the PV unit is displayed in Fig. 5. In addition to the timestamp, these include both the inverter data, e.g., AC/DC currents (A), AC/DC voltages (V), frequency (Hz), AC power output (W), and the data from the attached environmental sensor, e.g., module/ambient temperatures (°F), solar irradiance (W/m<sup>2</sup>) and wind speed (m/s).

```

Data retrieved from the smart inverter:
{ 'DAY_ENERGY': {'Unit': 'Wh', 'Value': 24683},
  'DeviceStatus': {'ErrorCode': 0, 'LEDColor': 2,
                   'LEDState': 0,
                   'MgmtTimerRemainingTime': -1,
                   'StateToReset': False,
                   'StatusCode': 7},
  'FAC': {'Unit': 'Hz', 'Value': 59.99},
  'IAC': {'Unit': 'A', 'Value': 28.78},
  'IDC': {'Unit': 'A', 'Value': 15.94},
  'PAC': {'Unit': 'W', 'Value': 6032},
  'TOTAL_ENERGY': {'Unit': 'Wh', 'Value': 11288108},
  'UAC': {'Unit': 'V', 'Value': 210.9},
  'UDC': {'Unit': 'V', 'Value': 392.9},
  'YEAR_ENERGY': {'Unit': 'Wh', 'Value': 1776127.62}}

Data retrieved from the attached sensor:
{ '0': {'Unit': '°F', 'Value': 59},
  '1': {'Unit': '°F', 'Value': 47},
  '2': {'Unit': 'W/m²', 'Value': 964},
  '4': {'Unit': 'm/s', 'Value': 1}}

Timestamp:
{ 'Timestamp': '2018-04-05T13:38:59-04:00' }

```

Fig. 5. Raw data retrieved from the smart inverter.

**B. P2P Trading Use Cases**

To allow P2P trading for exchange of excess solar energy, there are eight possible use cases, as summarized in Fig. 6.

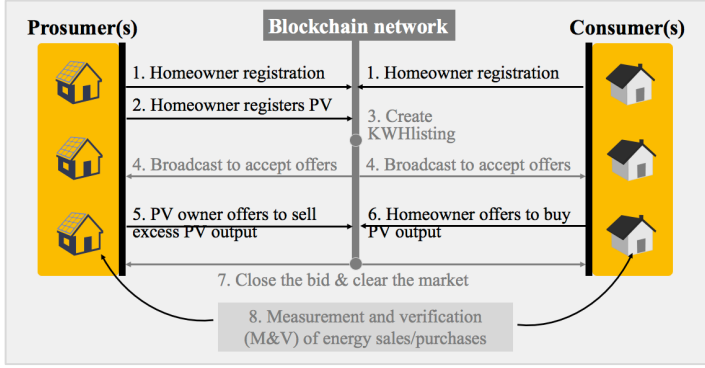


Fig. 6. Use cases illustrating transaction flows in the proposed platform (black text: participant generated actions; gray text: system generated actions).

*Use Case 1: Homeowner Registration* - A homeowner who would like to trade in the P2P trading perform must register as a participant in the blockchain network. To register, a homeowner submits the required information to the Hyperledger Composer REST server (using REST API) by running an instance of a Python code (Fig. 7). Executing this code registers 'seller1' to the deployed blockchain network. It specifies the homeowner's initial token balance, a unique homeowner ID (identified by an email address), as well as the first and last names.

```

url =
'http://xx.xx.xx.xxx:3000/api/org.acme.pv.auction.
Member?access_token=xxxxxxx'
payload = {
  "$class": "org.acme.pv.auction.Member",
  "balance": 200,
  "email": "seller1@vt.edu",
  "firstName": "seller1",
  "lastName": "seller1"}
headers = {'content-type': 'application/json'}
response = requests.post(url, data=json.dumps(payload),
headers=headers)

```

Fig. 7. A Python code for homeowner registration.

*Use Case 2: Homeowner Registers his/her PV (as an Asset)* - A homeowner with PV then registers his/her PV unit as an asset in the deployed blockchain network. To register, a homeowner specifies a unique PV ID and an owner ID identified by the email address. This was carried out through REST API in the similar fashion discussed in the Use Case 1, but with different URL and payload. See Fig. 8.

```

url =
'http://xx.xx.xx.xxx:3000/api/org.acme.pv.auction.
PV?access_token=xxxxxxx'
payload = {
  "$class": "org.acme.pv.auction.PV",
  "PVid": "seller1house ",
  "owner": "seller1@vt.edu"}
headers = {'content-type': 'application/json'}
response = requests.post(url, data=json.dumps(payload),
headers=headers)

```

Fig. 8. A Python code for PV registration.

*Use Case 3: Creating kWhlisting (Asset)* - To facilitate the hourly exchange of solar PV output, this study creates a blockchain asset called 'kWhlisting', which is created automatically every hour (by the system) to accept the offers to sell excess PV energy from prosumers and the bids to buy PV output from consumers. The system creates this asset hourly by sending the POST command shown in Fig. 9 to the REST server. As shown, this asset is identified by 'listingid', which is date and time of the exchange. The ACCEPT\_OFFERS state indicates that offers from sellers/buyers are now accepted. The 'pvoffers' and 'buyoffers' fields are initially blank, and intended to store offers from sellers and buyers (discussed in Use Cases 5&6, respectively).

```
url =
'http://xx.xx.xx.xxx:3000/api/org.acme.pv.auction.
kWhlisting?access_token=xxxxxx'
payload = {
"$class": "org.acme.pv.auction.kWhlisting",
"listingid": "21Mar18-13:00",
"KWhavailable": 0,
"state": "ACCEPT_OFFERS",
"pvoffers": [ ],
"buyoffers": [ ]}
headers = {'content-type': 'application/json'}
response = requests.post(url, data=json.dumps(payload),
headers=headers)
```

Fig. 9. A Python code to create the 'kWhlisting' asset.

*Use Case 4: Broadcast to Accept Offers* - At the time of auction, a message is automatically broadcasted to all participants (by the system), indicating that the offers are now accepted from both prosumers (for their PV energy sales) and consumers (for their energy purchases). This message is assumed to be broadcast one hour in advance of the actual transaction. That is, to enable a transaction at 13:00, this 'AcceptOfferBroadcast' message is sent out at 12:01. Once both prosumers and consumers receive this message, offers to sell PV energy are to be placed between 12:01 and 12:59, so are bids to buy PV energy. The 'AcceptOfferBroadcast' is associated with the 'listingid' (discussed in Use Case 3), which is the date and time of the exchange. In this case, the 'listingid' is 21Mar18-13:00.

*Use Case 5: PV Owner Offers to Sell Excess PV Output* - To allow each participant to listen to the 'AcceptOffer Broadcast' message, the instance of a Python code was configured to listen to the broadcast message from the Hyperledger-Composer Playground server using a WebSocket connection, as follows:

```
from websocket import create_connection
ws = create_connection("ws://xx.xx.xx.xxx:3000")
result = ws.recv()
```

Fig. 10. A Python code to listen to the broadcast from the REST server.

Once the 'AcceptOfferBroadcast' message is received, the owner may place an offer to sell his/her excess PV energy for the next hour. Each offer comprises: reservePrice (token/kWh),

kWhavailable (kWh), 'listingid' that this offer is responding to, and the unique PV ID (discussed in Use Case 2). To submit an offer, the required information is packaged in a payload and sent using the corresponding POST command to the REST server. Fig. 11 shows an example of the code to submit an offer to sell PV output. To determine 'kWhavailable' from PV for the next hour, a homeowner may run another piece of Python code to perform hour-ahead forecast of PV output based on the historical PV output data obtained from the smart inverter. Since this topic is not the focus of this paper, it is not included herein.

```
url =
'http://xx.xx.xx.xxx:3000/api/org.acme.pv.auction.
PVOffer?access_token=xxxxxx'
payload = {
"$class": "org.acme.pv.auction.PVOffer",
"reservePrice": 0,
"kWhavailable": 6,
"kWhlisting":
"resource:org.acme.pv.auction.kWhlisting#
21Mar18-13:00",
"pv": "resource:org.acme.pv.auction.Member#seller1
@vt.edu"}
headers = {'content-type': 'application/json'}
response = requests.post(url,
data=json.dumps(payload), headers=headers)
```

Fig. 11. A Python code to submit a PVOffer.

*Use Case 6: Homeowner Offers to Buy PV Output* - To place an offer, a homeowner specifies his/her bid price (token/kWh) and quantity (kWh) of PV output to purchase, as well as the 'listingid' that this offer is responding to. Each homeowner is specified by his/her unique ID, which is an email address. Fig. 12 shows an example of the code to submit a bid.

```
url =
'http://xx.xx.xx.xxx:3000/api/org.acme.pv.auction.
BUYOffer?access_token=xxxxxx'
payload = {
"$class": "org.acme.pv.auction.BUYOffer",
"BidPrice": 3,
"kWhQuantity": 5,
"listing":
"resource:org.acme.pv.auction.kWhlisting#
21Mar18-13:00",
"member":
"resource:org.acme.pv.auction.Member#buyer1
@vt.edu"}
headers = {'content-type': 'application/json'}
response = requests.post(url,
data=json.dumps(payload), headers=headers)
```

Fig. 12. A Python code to submit a BUYOffer.

*Use Case 7: Close the Bid and Clear the Market* - Closing the bid is carried out (by the system) hourly by calling the 'CloseBidding' transaction, associated with the 'listingid' to be closed. After the bid is closed, PV owners and homeowners are notified of their kWh sold/purchased amount. The tokens in their digital wallets are updated automatically to reflect the amount of sales (per the smart contract embedded in the 'logic.js' file). This completes the process and clears the market.

Use Case 8: M&V of Energy Sales/Purchases - At the last step, the data from smart meters are collected from each home for measurement & verification (M&V), and recorded on the blockchain network.

### C. Summary of the P2P Trading Implementation

The experiment was conducted using a case study of two sellers and three buyers. The participant registry as recorded in the Hyperledger Composer is shown in Table II. It is assumed that each seller and buyer has an initial balance of 200 tokens.

TABLE II.  
PARTICIPANT REGISTRY FOR ORG.ACME.PV.AUCTION.MEMBER

ID	Data (JSON)
seller1@vt.edu	{ "\$class": "org.acme.pv.auction.Member", "balance": 200, "email": "seller1@vt.edu", "firstName": "seller1", "lastName": "seller1" }
seller2@vt.edu	{ "\$class": "org.acme.pv.auction.Member", "balance": 200, "email": "seller2@vt.edu", "firstName": "seller2", "lastName": "seller2" }
buyer1@vt.edu	{ "\$class": "org.acme.pv.auction.Member", "balance": 200, "email": "buyer1@vt.edu", "firstName": "buyer1", "lastName": "buyer1" }
buyer2@vt.edu	{ "\$class": "org.acme.pv.auction.Member", "balance": 200, "email": "buyer2@vt.edu", "firstName": "buyer2", "lastName": "buyer2" }
buyer3@vt.edu	{ "\$class": "org.acme.pv.auction.Member", "balance": 200, "email": "buyer3@vt.edu", "firstName": "buyer3", "lastName": "buyer3" }

The PV asset registry is shown in Table III, indicating that each seller owns a PV unit.

TABLE III.  
ASSET REGISTRY FOR ORG.ACME.PV.AUCTION.PV

ID	Data (JSON)
seller1@vt.edu	{ "\$class": "org.acme.pv.auction.PV", "PVid": seller1house, "owner": "resource:org.acme.pv.auction.Member#seller1@vt.edu" }
seller2@vt.edu	{ "\$class": "org.acme.pv.auction.PV", "PVid": seller2house, "owner": "resource:org.acme.pv.auction.Member#seller2@vt.edu" }

Assuming that each of the two sellers and the three buyers submit an offer to the blockchain network in response to the 'AcceptOfferBroadcast' message with the 'listingid' of 21Mar18-13:00. The kWhlisting Asset registry is shown in Table IV.

The 'kWhlisting' asset with 'listingid' 21Mar18-13:00 indicates that the total kWh available for sales at 13:00 is 6+6=12kWh from both seller1 and seller2, and the total kWh to be purchased from three buyers is 5+3+1=9kWh. In this simplified scenario, it is assumed that each buyer offers the same bid price of 3 tokens/kWh, and no reserve price is set from the sellers. Once

the 'CloseBidding' transaction is called for 'listingid' 21Mar18-13:00, the exchange of excess PV generation is carried out based on the smart contract specified in the 'logic.js' file.

TABLE IV.  
ASSET REGISTRY FOR ORG.ACME.PV.AUCTION.KWHLISTING

ID	Data (JSON)
21Mar18-13:00	{ "class": "org.acme.pv.auction.kWhlisting", "listingid": "21Mar18-13:00", "state": "FOR_SALE", "pvoffers": [ { "\$class": "org.acme.pv.auction.PVOffer", "reservePrice": 0, "kWhavailable": 6, "listingid": "resource:org.acme.pv.auction.kWhlisting#21Mar18-13:00", "pv": "resource:org.acme.pv.auction.PV#seller1house", "transactionId": "5a47c319-9e82-4fc0-977a-42ddb25ee51c", "timestamp": "2018-03-21T12:50:06.040Z" }, { "\$class": "org.acme.pv.auction.PVOffer", "reservePrice": 0, "kWhavailable": 6, "listingid": "resource:org.acme.pv.auction.kWhlisting#21Mar18-13:00", "pv": "resource:org.acme.pv.auction.PV#seller2house", "transactionId": "b19e8faf-9760-491c-81c4-38a0219581e8", "timestamp": "2018-03-21T12:50:27.462Z" } ], "buyoffers": [ { "\$class": "org.acme.pv.auction.BUYOffer", "BidPrice": 3, "kWhQuantity": 5, "listingid": "resource:org.acme.pv.auction.kWhlisting#21Mar18-13:00", "member": "resource:org.acme.pv.auction.Member#buyer1@vt.edu", "transactionId": "ab754fa7-387c-4307-ae17-3602dfad5dc4", "timestamp": "2018-03-21T12:51:51.612Z" }, { "\$class": "org.acme.pv.auction.BUYOffer", "BidPrice": 3, "kWhQuantity": 3, "listingid": "resource:org.acme.pv.auction.kWhlisting#21Mar18-13:00", "member": "resource:org.acme.pv.auction.Member#buyer2@vt.edu", "transactionId": "1b8c2e65-36cb-4541-9f58-c2dbef989b10", "timestamp": "2018-03-21T12:51:28.912Z" }, { "\$class": "org.acme.pv.auction.BUYOffer", "BidPrice": 3, "kWhQuantity": 1, "listingid": "resource:org.acme.pv.auction.kWhlisting#21Mar18-13:00", "member": "resource:org.acme.pv.auction.Member#buyer3@vt.edu", "transactionId": "79537642-82c7-4d2a-8faf-f6638f1f9767", "timestamp": "2018-03-21T12:50:57.814Z" } ] }

The seller/buyer balances are updated, accordingly, which can be seen in the updated participant registry depicted in Table V. As shown, the balance of each homeowner has been updated correctly based on the deployed smart contract. That is, seller1 placed the offer earlier than seller2 (this is referring to the timestamp of each offer in Table IV). Hence, seller1 can sell the entire 6kWh of PV output, and receive the MCP of 3 tokens/kWh. As shown, the updated balance of seller1 is 200tokens + 3tokens/kWh\*6kWh = 218 tokens. Seller2 can only sell 3kWh of his PV output (since three buyers only need the total of 9kWh). Hence, the updated balance of seller2 is 200tokens + 3tokens/kWh\*3kWh = 209 tokens. Buyer1 purchases 5kWh at 3 tokens/kWh. Hence, his updated balance is 200tokens - 3tokens/kWh\*5kWh = 185 tokens. Similar calculations can be performed to figure out the updated balances of buyers 2 and 3.

TABLE V.  
PARTICIPANT REGISTRY FOR ORG.ACME.PV.AUCTION.MEMBER

ID	Data (JSON)
seller1@vt.edu	{"\$class": "org.acme.pv.auction.Member", "balance": <b>218</b> , "email": "seller1@vt.edu", "firstName": "seller1", "lastName": "seller1"}
seller2@vt.edu	{"\$class": "org.acme.pv.auction.Member", "balance": <b>209</b> , "email": "seller2@vt.edu", "firstName": "seller2", "lastName": "seller2"}
buyer1@vt.edu	{"\$class": "org.acme.pv.auction.Member", "balance": <b>185</b> , "email": "buyer1@vt.edu", "firstName": "buyer1", "lastName": "buyer1"}
buyer2@vt.edu	{"\$class": "org.acme.pv.auction.Member", "balance": <b>191</b> , "email": "buyer2@vt.edu", "firstName": "buyer2", "lastName": "buyer2"}
buyer3@vt.edu	{"\$class": "org.acme.pv.auction.Member", "balance": <b>197</b> , "email": "buyer3@vt.edu", "firstName": "buyer3", "lastName": "buyer3"}

## V. CONCLUSION

With an increasing number of rooftop solar PV, the blockchain technology can play a crucial role in supporting future complex P2P energy trading applications. This paper presents an initial proof-of-concept of a blockchain-based P2P energy trading platform for exchange of solar energy. It defines participants, assets, transactions, as well as the smart contract, and discusses the implementation of the proposed platform using the open-source Hyperledger fabric blockchain framework. Description is given in details on how the Hyperledger Composer Environment was set up, together with the Hyperledger Composer REST server, participant identities and wallets. Use cases are discussed that allow P2P trading for exchange of excess solar energy. Then, an experiment was set up in a laboratory environment to demonstrate the use of the proposed blockchain-based network in supporting P2P trading applications. Overall, this paper serves as a basis describing how one could leverage a blockchain technology to allow record keeping of all complex transactions and operational data in the context of exchanging excess solar PV output. It enables transactions for exchanging electricity (kWh) generated locally among different participants, which is the key to enable prosumers to transact in a viable market. It is expected that this paper will benefit researchers and utility engineers who would like to develop, deploy and experiment on a blockchain network for P2P trading of solar energy.

## VI. REFERENCES

- [1] Pyper, J., "The US solar market is now 1 million installations strong", GTM, April 2016 [Online]. Available: <https://www.greentechmedia.com/articles/read/the-u-s-solar-market-now-one-million-installations-strong#gs.ze2WCvM>. Retrieved: February 2018.
- [2] Solar Energy Industries Association (SEIA), "Solar industry data" [Online]. Available: <https://www.seia.org/solar-industry-data>. Retrieved: February 2018.
- [3] IEEE Blockchain [Online]. Available: <https://blockchain.ieee.org>. Retrieved: November 2017.

- [4] C. Zhang, J. Wu, C. Long and M. Cheng, "Review of Existing Peer-to-Peer Energy Trading Projects", *Energy Procedia*, vol. 105, pp. 2563-2568, 2017.
- [5] A. Goranović, M. Meisel, L. Fotiadis, S. Wilker, A. Treytl and T. Sauter, "Blockchain applications in microgrids: an overview of current projects and concepts," *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, 2017, pp. 6153-6158.
- [6] PowerLedger [online]. Available: <https://powerledger.io/>. Retrieved: April, 2018.
- [7] PowerLedger whitepaper [Online]. Available: <https://powerledger.io/media/Power-Ledger-Whitepaper-v8.pdf>. Retrieved: April, 2018.
- [8] Brooklyn Microgrid [Online]. Available: <http://www.brooklynmicrogrid.com>. Retrieved: April, 2018.
- [9] DAJIE!, May 2017, [online] Available: <https://www.dajie.eu>. Retrieved: April, 2018.
- [10] Shareandcharge, [online] Available: <https://shareandcharge.com/>. Retrieved: April, 2018.
- [11] Ethereum blockchain app platform [Online]. Available: <https://www.ethereum.org>. Retrieved: April, 2018.
- [12] NRGooin - Smart Contract for green energy, [online] Available: <http://nrgooin.org>. Retrieved: April, 2018.
- [13] Solar Coin [Online]. Available: <https://solarcoin.org>. Retrieved: Nov 2017.
- [14] "The Silicon Based Economy - financing solar cells with Bitcoin", *The Sun Exchange*, Jan 2017 [online]. Available: <https://thesunexchange.com/silicon-based-economy-financing-solar-cells-bitcoin>. Retrieved: April, 2018.
- [15] Electron project [Online]. Available: <http://www.electron.org.uk>. Retrieved: April, 2018.
- [16] M. Merz, "The enerchain Project", *enerchain*, [online] Available: <https://enerchain.ponton.de/>. [accessed: 2018-04-03].
- [17] M. Castillo, "Nasdaq Explores How Blockchain Could Fuel Solar Energy Market", May 2016 [Online]. Available: <https://www.coindesk.com/nasdaq-blockchain-solar-power-market/>. Retrieved: April, 2018.
- [18] Slock.it [Online]. Available: <https://slock.it>. Retrieved: April, 2018.
- [19] WePower [Online]. Available: <https://wepower.network>. Retrieved: April, 2018.
- [20] E. Mengelkamp, B. Notheisen, C. Beer, D. Dauer and C. Weinhard, "A blockchain-based smart grid: towards sustainable local energy markets", *Comput Sci Res Dev.*, vol. 33, pp. 207-214, 2018.
- [21] P. Chankook and Y. Taeseok, "Comparative review and discussion on P2P electricity trading". *Energy Procedia*, vol. 128, pp.3-9, 2017.
- [22] A. Hahn, R. Singh, C.C. Liu and S. Chen, "Smart contract-based campus demonstration of decentralized transactive energy auctions", In *Proc. IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC., pp. 1-5, 2017.
- [23] E. Mengelkamp, J. Gärtner, 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.
- [24] J. Hwang, M. Choi, T. Lee, S. Jeon, S. Kim, S. Park and S. Park, "Energy Prosumer Business Model Using Blockchain System to Ensure Transparency and Safety", *Energy Procedia*, vol. 141, pp. 194-198, 2017.
- [25] E. R. Sanseverino, M. L. D. Silvestre, P. Gallo, G. Zizzo and M. Ippolito, "The Blockchain in Microgrids for Transacting Energy and Attributing Losses," *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Exeter, 2017, pp. 925-930.
- [26] K.N. Khaqqia, J.J. Sikorskib, K. Hadinotoa and M. Kraft, "Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application", *Applied Energy*, vol. 209, pp. 8-19, January 2018.
- [27] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," in *IEEE Access*, vol. 4, pp. 2292-2303, 2016.
- [28] Hyperledger [Online]. Available: <https://hyperledger.github.io>. Retrieved: April, 2018.
- [29] Hyperledger Composer [Online]. Available: <https://hyperledger.github.io/composer/latest/>. Retrieved: April, 2018.
- [30] Installing pre-requisites [Online]. Available: <https://hyperledger.github.io/composer/installing/installing-prereqs.html>. Retrieved: Apr, 2018.
- [31] Installing the development environment [Online]. Available: <https://hyperledger.github.io/composer/installing/development-tools.html>. Retrieved: April, 2018.