This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

Aggregation and remuneration in Demand-Response with a blockchain-based framework

M.L. Di Silvestre, P. Gallo, E. Riva Sanseverino, G. Sciumè and G. Zizzo, *Senior Member, IEEE,*

***Abstract*—This paper describes the possibility to use the blockchain technology for load and generation aggregation in a new distributed Demand Response (DR) service and customers remuneration system. The blockchain technology and the use of smart contracts for DR allow the creation of a distributed system in which customers can communicate directly, in a transparent, secure and traceable way, with the grid operator to provide their flexibility. In this paper, the DR problem formulation takes into account several aspects, which are periodically executed. First, the blockchain records customers' energy consumption or production, then, the smart contract starts calculating the baseline and the potential support provided by each customer to fulfill the requested load adaptation. Customers' availability for generation and load profile modulation is also taken into account, as well as their privacy and an updated definition of the roles of grid and market operators in a new Demand-Response scenario supported by the blockchain technology. The blockchain used is Hyperledger Fabric, since it turned to be flexible for smart contracts implementation while supporting multi-tenancy. Results show the possibility to successfully apply the blockchain technology to this particular topic, even considering privacy-preserving issues.**

***Index Terms*—blockchain; transactive energy; microgrids; energy market; peer to peer; smart contract; demand response.**

LIST OF SYMBOLS

α	learning rate
β	weight of the terms of the availability profile
ϕ	overall desired load reduction
ϕ_j	overall desired load reduction at the j-th iteration of the smart contract
σ	sensitivity of the smart meter
c	customer's identifier
CL	cluster of customers
CL_j	cluster of customers at the j-th iteration of the smart contract
h	hour of the day
i	index of the DR event
r	remuneration
$\mathbf{A}$	vector of filtered availability profile for the 24 hours
$\mathbf{A}^*$	vector of current availability profile for the 24 hours
$\mathbf{B}^{(c)}$	baseline vector of the customer c
$\mathbf{B}^{(CL)}$	baseline vector of the cluster CL
$A_{h,i}$	availability at hour h during the DR event i
ΔP	load reduction performed by the customer
$\Delta P_{B,h,i}$	difference between the baseline and the target consumption at hour h for the i-th DR event
ΔP_B	difference between the baseline and the target consumption
$\Delta P_{DR,h,i}$	difference between the baseline and the target consumption for a customer at hour h for DR event i
ΔP_{DR}	difference between the baseline and the target consumption for a customer
$\bar{P}^{(c)}_{B,h}$	baseline's value of the customer c at hour h
P	measured load during a DR event
$P^{(CL)}_B$	baseline's value of the aggregated load of the cluster
$P_{h,i}$	measured load at hour h during the i-th DR event
$P_{t,h,i}$	requested target load during the i-th DR event at hour h
P_t	requested target load during a DR event
$\Psi^{(c)}$	fraction of the load reduction to be required from a customer
$\mathcal{H}$	Heaviside function

(R2.6),(R3.1)

I. INTRODUCTION

BLOCKCHAIN capabilities are evolving and Smart contracts technologies are peaking the Gartner hype cycle for blockchain business in 2019 [1]. A whole set of industry-related blockchain applications is now appearing [2], [3]. Moreover, if we observe the position of blockchain technology for the utility industry in 2019, it is still on the rise, witnessing the big expectations as a critical technology for delivering reliable, affordable and ubiquitous commodity services in the power systems field. Blockchain has many applications in the energy field mostly aiming at energy trading between peers [4]–[7] such as producers or prosumers [8] or electric vehicles [9], [10]. Many recent papers have presented very complete surveys on energy blockchain and transactive energy [4], [7], [11]–[16]; however, few papers have considered in detail the application of the blockchain technology to the delivery of electrical load management services. One of the first papers on the topic [17], proposes the use of decentralized blockchain mechanisms for delivering transparent, secure, reliable, and timely energy flexibility services. In this way, the adaptation of energy demand profiles of electricity prosumers is transparent to all the stakeholders involved in the flexibility market (Distribution System Operators primarily, retailers, aggregators, etc...). In [17], the Ethereum platform is used for handling smart contract's execution. However, it seems that no consideration is made about energy consumption and costs associated with the execution of smart contracts in this permissionless environment. Moreover, privacy-preserving mechanisms need to be implemented, since electrical consumption data are considered as personal data. According to the General Data Protection Regulation [18], they cannot be shared even under

Manuscript received Month day, year; revised month day, year.

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

pseudonymization. The possibility offered by the Hyperledger fabric of selective transparency on data allows, on one hand, transparency on what is relevant for smart contracts implementation and, on the other, sufficient privacy preservation for personal data. Another interesting paper [19] tried to combine blockchain technology with the interoperability of the OpenADR protocol, with a new paradigm for Demand Response (DR). Even in this case, the Ethereum blockchain is used and the problem of handling privacy is not completely solved. The work in [20] proposed an energy sharing-model with price-based DR where a dynamical internal pricing model is formulated for the operation of an energy-sharing zone.

In [21] the authors proposed a P2P aggregation framework for providing ancillary services to the utility where blockchain can be used for managing the flexibility exchange. In [22], a distributed direct load control scheme for large-scale residential DR was proposed, built on a two-layer communication-based control architecture where an energy management controller uses wireless links to schedule operation of appliances according to a local power consumption target.

To the knowledge of the authors, none of the papers on blockchain and DR deals with a transparent evaluation of the Customer Baseline Load [23] (CBL) for each customer with smart contracts, as well as with the customer's remuneration depending on deviation from the CBL. The CBL describes the typical customer's consumption profile and permits to evaluate if changes occur in response to a DR program. The latter is deduced based on measurements available from smart metering devices on a defined number of days. The baseline should be recalculated along time, as the consumption habits may change in the long term. The CBL estimation is a strategic issue in the successful implementation of DR programs and in designing compensation mechanisms. Indeed, the difference between the CBL and the actual consumption represents the customers' performance under the DR program and is the reference to design the economic compensation mechanism. The accurate estimation of the CBL is thus critical to the success of DR programs because it involves the interests of multiple stakeholders including utilities and customers. For this reason, the literature has extensively treated the problem of CBL estimation. For example, the work in [23] proposes an estimation approach of CBL for residential customers based on clustering. The work in [24] proposes an error analysis of the same CBL calculation methods considered in [23] to determine which methods are best suited for residential customers. The 'HighXofY' method [25], [26] is currently adopted by the New York Independent System Operator (NYISO) and provides the average of the consumption of the highest X consumption days within a range of Y non-DR days preceding the day under analysis. This method is based on the hypothesis that the load profile of an individual customer during the weekdays is different from the load profile during the weekends, as proposed in [23] and in [24]. In [27] the authors analyze the DR baselines for residential customers including the impact of the baselines on the profit of all parties concerned, customers and company, proposing a novel yet relatively simple baseline method called ”LowXofY”. This method has a more negative bias than baselines evaluated with the other methods, but it is more accurate. In [23] the authors improve the accuracy of CBL estimation for incentive-based DR programs through a CBL residential estimation that uses the Synchronous Pattern Matching principle. This new method, compared to the non-synchronous estimation methods, presents significantly better overall performance. In the experiments here proposed, the authors use the 'HighXofY' method for assessing the CBL, due to the continuous update of the CBL and the ease of implementation. **(R2.1)**

This paper proposes a blockchain-based framework for implementing DR where the smart meter provides data of load or generation and such data is loaded on the blockchain through a client application. Then, a dedicated smart contract computes the baselines using the 'HighXofY' algorithm and adds them to the ledger. The day before the delivery of the service, the grid operators (DSO/TSO) publish the request of load reduction to avoid congestion in the network, in terms of time window and reduction amount. Under the hypothesis that all customers take part in the DR program, a dedicated smart contract distributes to the customers the total load reduction communicated by the grid operator, taking into account the availability of each customer to modulate his/her generation or load. After the event, customers can invoke the smart contract to check how much their consumption has been compliant with the grid operator's request. The smart contract remunerates the customers with energy tokens, according to a parabolic remuneration function prizing those customers that better met the grid operator's request.

The proposed approach can be useful in various applications where a bottom-up or peer-to-peer aggregation for providing flexibility to the grid operators can be implemented, like in the scenarios proposed in [21] and [28]. In this latter case, in the absence of a third-party aggregator, the approach can be applied using a Virtual Aggregation Environment, as described in [21].

The main contributions of this paper reside in the experimentation, in a laboratory, of using smart contracts for aggregating loads instead of private applications owned by different parties. As a result, the entire aggregation for demand response can be disintermediated. The main functions executed by the implemented smart contracts for this work are:

- CBL calculation;
- Calculation of each customer's flexibility through an indicator called "*Availability Profile*";
- Notification of DR events;
- Verification of compliance to the grid operator's request;
- Assignment of load reduction and remuneration to the customers.

(R2.3)

II. DR PROBLEM FORMULATION

The DR formulation takes into account a scenario where the grid operator asks the customers to modify their loads to meet the technical requirements of the grid. The grid operator sends DR signals to change the loads, in the range of the available flexibility obtained as the sum of the flexible powers declared by the customers. Each customer has a different inclination

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

(availability) to follow the communicated indications. Indeed, loads adapt according to an *availability function* that analytically implements the customer's willingness to participate in DR events: some customers are flexible enough to change their typical load profile, others do not want or cannot adapt. Customers' typical consumption profiles are the baselines, they are references to compute the actual customers' adaptation to DR events. In the rest of the paper, all customers of the cluster are assumed taking part in DR events, which are considered composed by the following steps, which are run periodically:

1) Smart meters record the customers' load consumption;
2) The customers' client applications send to the blockchain the new metered data;
3) Before the communication of a new DR event, the grid operator triggers the baselines and availability profiles calculation functions of the smart contract;
4) The grid operator notifies the DR event on the blockchain: day, time window and total load reduction/increase;
5) The smart contract, considering the baselines and the availability profiles evaluated, distributes the overall request to the customers;
6) The day of the DR event, each customer reads from the blockchain the desired load reduction evaluated by the smart contract in order to satisfy the grid operator's request;
7) The DR event takes place;
8) The customers' load consumption during the DR event is recorded on the blockchain as explained in steps 1) and 2);
9) The day after the DR event, the market operator triggers the smart contract to check if customers' consumption has been compliant with the grid operator's request.
10) If the customers' consumption has been compliant with the request, the smart contract remunerates them with energy-tokens.**(R3.4)**

The baseline of a customer c consists of a vector of typical power consumption in 24 hours, whose general expression is given below in (1).

$$\mathbf{B}^{(c)} = \left[\bar{P}^{(c)}_{B,1}, \bar{P}^{(c)}_{B,2}, ..., \bar{P}^{(c)}_{B,h}, ..., \bar{P}^{(c)}_{B,24}\right] \tag{1}$$

The horizontal bar above the symbols of power consumption $\bar{P}^{(c)}_{B,h}$ indicates that load values related to a specific hour h of the day d are dynamically averaged over multiple days according to the following formula:

$$\bar{P}^{(c)}_{B,h} = \frac{1}{X} \sum_{j \in High(X,Y,d)} P^{(c)}_{B,h,j} \quad \forall\, h \in \{1, 2..., 24\} \tag{2}$$

This averaging method for CBL calculation is called "High-XofY" [24]. The left term in (2) is the value of the baseline at hour h by averaging the power of the X days with the highest consumption within the Y non-DR days preceding day d. This averaging methodology works on a set of the Y days that do not include DR or curtailment programs, holidays, etc. Please refer to [25] for further details. **(R2.1)** In the rest of this paper, the day d is not explicitly indicated as it is the day on which the DR event is notified to customers, which is assumed is the day before the DR event. The baseline is calculated over the Y preceding days belonging to the same group (weekdays, holidays, etc.) of homogeneous consumption. The sliding window over the last Y days permits to take into account seasonality.

The vector sum of all the baselines for the customers belonging to a cluster CL is the typical power profile of the aggregated load, as indicated in (3).

$$\mathbf{B}^{(CL)} = \sum_{c \in CL} \mathbf{B}^{(c)} \tag{3}$$

When the grid operator communicates a load reduction for a given hour h, the customer's client application runs the smart contract to check what is the assigned load reduction during hour h leading to the maximum remuneration.

This assigned load reduction for each customer is calculated by the smart contract taking into account the *"Availability Profile"* (A) of the customer, which represents the customer's inclination to respond to the request by the grid operator. This quality figure shows customers' performance as the sum of two terms: the first provides information on how difficult is the reduction requested by the operator, named *difficulty*; the second provides how good is the customer's compliance while following DR indications, named *compliance*. Additionally, $\boldsymbol{A}$ evolves at any new DR event. For ease of notation, the superscript (c) has been omitted in all subsequent formulas. The $\boldsymbol{A}$ of a generic customer is a vector of 24 elements, as many as the hours in a day. The 24 elements of the vector $\boldsymbol{A}$ represent the availability to modify the load at hour h, during the *i-th* DR event. This value is composed of two terms weighted by β and $1-\beta$, both ranging between 0 and 1, as in (4).

$$A^{*}_{h,i} = \beta\, \frac{\Delta P_{B,h,i}}{\bar{P}_{B,h}} + (1-\beta)\, \frac{\sigma}{\sigma + \Delta P_{DR,h,i}} \tag{4}$$

(R3.2)

In this expression, $\Delta P_{B,h,i}$ is the difference between the baseline $\bar{P}_{B,h}$ at hour h and the target consumption $P_{t,h,i}$ at the same hour; σ is the sensitivity of the smart meter (typical value is taken as 30W, considering an accuracy of 1). $\Delta P_{DR,h,i}$ is the absolute difference between the customer's load profile measured during the DR event and the target load profile. The two terms of the *Availability Profile* are both normalized as $\Delta P_{B,h,i}$ / $\bar{P}_{B,h} \in (0, 1]$ and $\sigma/(\sigma+\Delta P_{DR,h,i}) \in (0, 1]$. Both terms are calculated at the *i-th* DR event occurring at hour h. The first term accounts for what fraction of the baseline has been requested to be changed, $\Delta P_{B,h,i}$=$|\bar{P}_{B,h}$-$P_{t,h,i}|$ (See Fig. 1). The bigger is the first term and the higher is the effort requested to the customer. The second term accounts for the actual "capacity" of the customer, $\Delta P_{DR,h,i}$ = $|P_{h,i}$-$P_{t,h,i}|$, defined as the difference between the measured load profile $P_{h,i}$ and the requested target load $P_{t,h,i}$ for the customer c at the hour h. This term measures the customer's capability to 'follow' the directions given by the grid operator. The smaller is $\Delta P_{DR,h,i}$ and the more compliant to the grid operator's request is the customer's load profile. In (4) $\Delta P_{DR,h,i}$ is compared to σ, which is the sensitivity of the smart meter, so when $\Delta P_{DR,h,i}$ is close to zero, the second term is close to 1.

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

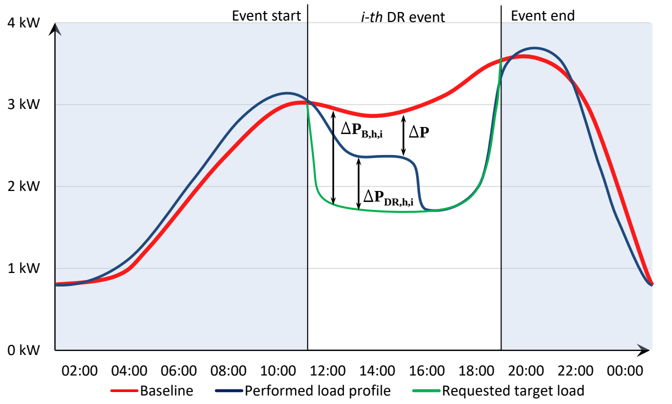


Fig. 1: Quantities involved in an exemplary Demand Response event.

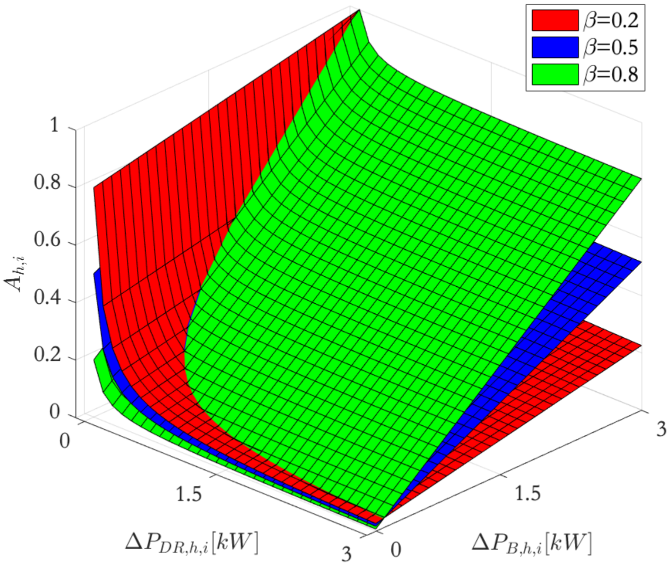


Fig. 2: Availability of a customer at hour h for the DR event i, when $\beta \in \{0.2, 0.5, 0.8\}$

In (4), the value of $\beta \in [0, 1]$ defines which of the two terms is the most important to evaluate customer behaviour. On one hand, β higher than 0.5 will prize those customers who had been requested more effort in the adaptation (first term), even if they have low compliance (second term). On the other hand, β lower than 0.5 will prize customers that follow precisely the target profile (second term), even if it does not require such an effort (first term). We propose to set β = 0.2 as the second compliance term has more stability effects on the system than the first on difficulty. In fact, in the long run, the system will require greater effort to those that performed better, therefore their difficulty is expected to increase in subsequent DR events. However, when the required effort increases too much, the customer will presumably not be able to respond adequately. As a consequence, the availability profile will reduce in a closed-loop control. Instead, if the weight of the difficulty term is dominant ($\beta > 0.5$), more reduction will be assigned to those that were requested more, without caring if they are able to follow the indications. Fig. 2 reports how the availability profile depends on $\Delta P_{B,h,i}$, and $\Delta P_{DR,h,i}$, which are hypothesized belonging to domestic users whose residential committed power is 3 kW. The figure shows how the $A_{h,i}$ values, hosted in the z-axis, are between zero and one in the three cases discussed above for $\beta \in \{0.2, 0.5, 0.8\}$. **(R3.2)**

According to Fig. 1, from 11:00 to 16:00, the customer did not fully satisfy the reduction requested by the grid operator ($\Delta P_{DR,h,i} \neq 0$), instead, he fully satisfied the request from 16:00 to 19:00 ($\Delta P_{DR,h,i} = 0$). Therefore, if the difficulty to accomplish the request for the customer is high, even if he was not compliant, the availability will tend to be high anyway. If the difficulty to accomplish the request is low, the not compliance is more severely considered and the availability is low. Of course the two terms are then weighted using β.

The *Availability* is thus recalculated at each DR event but to account for past events and filter out steep behavioural variations, a memory term is considered. The value of *Availability* becomes more and more defined as the number of DR events to which the user takes part increases, using the learning rate α. Such a real parameter takes a value lower than 1. As a result, the *Availability* of a generic customer calculated after the *i-th* DR event is expressed as follows:

$$A_{h,i} = \alpha \cdot A^*_{h,i} + (1 - \alpha) \cdot A_{h,i-1} \qquad (5)$$

where:

- i indicates the last DR event;
- $\alpha \in$ [0,1] is the learning rate for the time-based adaptation of the *Availability Profile* of the customer.

In this case, α has been set to 0.2 to filter out steep behavioural variations of users.**(R2.8)** As explained in the following paragraphs, the Availability is used to calculate the fraction of load reduction required to each customer belonging to a cluster *CL* (we will consider only a load reduction and not an increase to simplify the explanation). The easiest way to attribute a load reduction to a single customer belonging to a cluster is to share it proportionally to the ratio (baseline)/(baseline of the cluster) at a given hour. However, this method do not account for the availability of a customer to respond to a given input from the grid operator. On the other hand, altering the strict proportional partition of the load reduction leads to a request that can be higher or lower than the overall request. For this reason, while the authors propose to include the availability in the assessment of the fraction of load reduction to be given to each customer, they also propose to correct iteratively the assignment so as to reach a good match between assignment and request. In this way, the whole DR technique will gain reliability and precision. If ϕ is the overall desired load reduction communicated by the grid operator for a given hour h for the *i+1* DR event, the smart contract will distribute this overall request to the customers belonging to the cluster *CL* considering the $A_{h,i}$ of the customers, and using the following iterative algorithm that at any step *j* assigns a fraction of the residual load reduction, $\Psi^{(c)}_{h,j}$, as follows:

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

$$\Psi^{(c)}_{h,j} = \phi_j \cdot \frac{(1+A_{h,i})}{2} \cdot \frac{\bar{P}^{(c)}_{B,h}}{P^{(CL_j)}_{B,h}} \tag{6}$$

where $c \in CL_j$ and $CL_j \subseteq CL_{j-i} \subseteq \cdots \subseteq CL_0 = CL$:

$$\phi_j = \begin{cases} \phi & \text{if } j = 0, \\ \phi - \sum_{c \in CL_j} \sum_{k=1}^{j-1} \Psi^{(c)}_{h,k} & \text{otherwise.} \end{cases} \tag{7}$$

The total load reduction requested to a customer at step j is, therefore, expressed by:

$$\Delta P^{(c)}_{B,h,j} = \sum_{k=1}^{j} \Psi^{(c)}_{h,k} \tag{8}$$

In (6), the iterative algorithm, at any step, assigns to the cluster a fraction of the residual reduction load between 0.5 and 1, as $A_{h,i} \in [0,1]$. The cluster CL_j changes at any step j of the algorithm; in fact, those customers that have already been requested to reduce their loads to zero do not participate in the next assignment steps, therefore $CL_j = \{c \mid \Delta P^{(c)}_{B,h,j} < \bar{P}^{(c)}_{B,h}\}$. The exit condition of the iterative algorithm is that the residual reduction is lower than a threshold. As reported in (4), if the customer is reliable and/or has a very high potential to respond to the request, then, he/she will be called to a higher effort during the next DR event. The *Availability* is initialized to 0.5 when the user has not yet participated in any DR event at the specific hour h, assuming a neutral behavior. Then, at any DR event, the *Availability* will be updated according to (5) and the smart contract will assign a load variation according to (6). In (6) the alteration of both the individual and cluster baselines are functional to the update of the assignment of the load reduction to the customers, according to their availability. **(R3.7)**

Currently, in the management of DR events, the grid operator communicates to the aggregator the request for load modulation in a given area of the electrical system. The participation of consumers/producers in the DR service takes place through their aggregation in virtual units, set up and managed by the aggregator or Balancing Service Provider (BSP), that is responsible for responding to modulation orders given by the grid operators. The aggregator or BSP that knows the total capacity of customers of that area, who participate in the DR service, responds to these orders by requesting, in compliance with the constraints and requirements necessary for the aggregated resources, the modulation of the load in the virtual load unit [29]. The price that the grid operator will pay to the aggregator for its service is determined through an auction between them; then, at the end of the DR event, the aggregator pays customers according to the agreed remuneration system [30]. The day in which the DR service takes place, the customers' consumption is recorded by the smart meters. At the end of the day, the smart contract remunerates customers.

Customers' remuneration is one of the most important aspects of DR programs. Indeed, remuneration is vital for maintaining a DR program along time. Various remuneration schemes are present in the literature. Some methods provide a constant reward for each unity of energy consumption that has been modified by the customer, others consider the remuneration as a function of the change in the customer's profile [29]. Still, the remuneration can change with the time of the day (as in Time-of-use schemes) or with the kind of customer. As an example, in [14], an optimal bidding strategy is proposed where a Load Aggregator manages the flexible resources of the customers in order to maximize the additional profit while bidding in the Day-Ahead market. At the same time, the Aggregator remunerates the customers considering both the deviation from the customers' baseline and the reduced power. In [31] the authors consider different remunerations for five categories of customers. In [32] the authors propose a remuneration scheme considering an incentive price dependent on the customer's power modification due to the participation in a DR program, and the differences between the quantity of energy bought by the Utility in the Spot market and the retail price for the customers, so as to limit the loss for the Utility. In [33], instead, the objective to maximize while remunerating the customers is the profit for the Aggregator. Finally, in [34] a positive/negative (limited) quantity is added to the retail price in each period of a day in order to encourage load shifting from a price period to another. **(R2.2),(R2.3),(R3.5)** In this paper, a quadratic remuneration function is used. The latter is usually adopted for economical evaluations in the electric energy field [30], and for the DR event is expressed as follows:

$$r = \sum_{h=1}^{24} \mathcal{H}(\bar{P}_{B,h} - P_h) \cdot (\bar{P}_{B,h} - P_h)^2 \tag{9}$$

where $\mathcal{H}$ is the Heaviside function; it is 1 when the argument is positive, and 0 otherwise, permitting to remunerate only load reductions and neglecting load increments. **(R3.6)** In (9), $\bar{P}_{B,h}$ is the baseline value at hour h, P_h is the measured load profile at hour h and their difference is representative of customer response during the DR event (see ΔP in Fig. 1). In our experiments, the operator requests load reductions and only those consumers that reduce their load receive a reward, according to a parabolic law. Additionally, the proposed approach may be modified using different remuneration functions (e.g. depending on the season, the day of the week of the DR event, etc.) **(R3.6)**

III. The Blockchain Platform for DR

The proposed blockchain platform addresses efficiently the two most relevant issues that prevent applying the blockchain to DR programs: privacy issues and a the presence of a large amount of measured data to be stored and shared in the blockchain.

A. Privacy issues

Hyperledger Fabric consists of three types of nodes: Orderers (responsible for establishing consensus), Peers (host the ledger and smart contracts, here called chain codes) and Clients (used for submitting transactions). Transactions are implemented by interactions with the peer nodes hosting the ledger (where data reside) and the chaincodes (smart

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

contracts). Interactions, in turn, can be reading data from the ledger, writing data to the ledger, running chain codes and writing the results on the ledger. All members of Hyperledger Fabric blockchain are grouped into organizations and multiple organizations can group into consortia. An organization can host more Peer and Client nodes. The transactions are submitted by the Client nodes. The peer nodes host the chain codes and the ledger itself. In our application for DR, transactions are submitted by the grid operator, market operator or by the customers. Therefore, the validation process must guarantee that organizations do not misbehave. There is no central authority for validation but a policy called "Endorsement Policy". Endorser peers, which are the peers in which the chain code is installed, do a similar task. They make sure that before updating an asset, namely before updating the blockchain and its content, its state is correct. Once a transaction is endorsed by the relevant participants, Orderer Service Nodes comes into play. At this time, the client broadcasts endorsed (signed) transactions to Orderer(s). Orderers verify that the client has an appropriate role, which is required to modify the ledger and in case it is confirmed, collect transactions into blocks and broadcast them to the channel when blocks reach a defined size or after a specific amount of time [35]. In the proposed application, the authors consider two organizations grouped in a consortium called DR Consortium. The first organization (Org1) is composed of grid and market operators, while the second (Org2) is composed of market operators and customers (see Fig. 3). For private communication between members of a consortium, Fabric uses the channels, communication mechanisms among peers that allow data isolation. Through the Client, the members of the DR Consortium can execute transactions interacting with the smart contract that is installed on all the peers. The communication takes place through communication channels so that members of the consortium can communicate with each other. The identities of all the members of the network are encapsulated in X.509 digital certificates, dispensed by Certificate Authorities (CA). These certificates are verified by Membership Service Providers (MSPs), which define the rules that govern valid identities for each organization. Generally, every organization has its MSP and CA [36]. In this application, two MSP were considered, one for Org1 and one for Org2. So the MSP of Org1 is managed by the grid operator and is needed to verify the identity of the market operators, while the MSP of Org2 is managed by the market operator and is needed to verify the identities of the customers. An important feature of the digital identity is the possibility to include inside it some additional attributes, which fabric uses to determine permissions to use the different functions implemented by the smart contract. In the proposed scenario, the CA of the Org1 includes the ID of the market operator inside his digital identity and the CA of Org2 includes the smart meter ID inside the digital identity of each customer. Thanks to this, it is possible to manage the permissions for the use of smart contract functions and privacy among the various network users. In fact, each customer can read from the blockchain only those data that concern him, such as his own baseline, his own consumption or earned tokens, but he cannot read the metered data related to another customer. Indeed at that time, if he tries to make a query about these data he will get an error signal since only the ID of his smart meter is present in his digital identity. Fig. 3 shows the architecture of the proposed blockchain-based platform for DR.

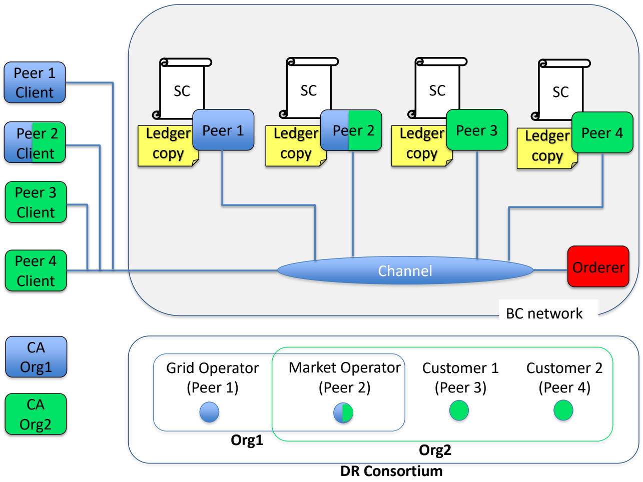


Fig. 3: DR Blockchain-based network.

B. Managing large amounts of data for DR

In the proposed architecture, it is possible to adopt two different approaches to solve the problem of managing large amounts of data. The first is to create different channels for sharing data. One channel for each grid operator-market operator-customer group in which the consumption data are shared, used for the baseline calculation, and a general channel to which all the customers of the network take part where the data useful for calculating remuneration and the remuneration itself are visible. The second is to create a single channel and use private data collection. This feature of Fabric allows a defined subset of organizations on a channel to support, commit, or query private data without having to create a separate channel. A private data collection is a combination of two elements:

1) Current private data: sent peer-to-peer via a gossip protocol only to organizations authorized to view them. This data is stored in a private database on the peer.
2) A hash of such data: which is approved, ordered and written in the ledger of each peer on the channel. The hash serves as proof of the transaction and is used for state validation and can be used for control purposes.

Furthermore, there is the possibility of automatically removing a part of the data that turns out to be useless after a certain period of time. In the case considered, for example, it is possible to delete the data used to calculate the baseline, after the latter has been memorized on the blockchain state [37].

In this work, the second solution has been adopted.

IV. Application

Hyperledger Fabric is a blockchain with highly modular and configurable architecture, which allows versatility and

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

optimization for a wide range of use cases. Fabric is the first distributed accounting platform to support smart contracts written in generic programming languages, such as Java, Go and Node.js. The Fabric platform is permissioned, which means that, unlike a public network without authorization, the participants are known to each other rather than anonymous.

One of the most important properties of the platform is the possibility to choose the consensus protocol. For example, if may be implemented within a single company, can be a proof-of-work, proof-of-stake, a Byzantine Fault Tolerance (BFT), etc. In addition, Fabric may use consensus protocols that do not require a native crypto-currency to incentivize expensive mining activities or to fuel the execution of smart contracts [38]. The absence of cryptographic mining operations allows the platform to be distributed at the same operating cost as any other distributed system.

The combination of these features makes Fabric a highly performing platform in terms of transaction processing and transaction confirmation latency and allows privacy, transaction confidentiality and the implementation of smart contracts [39]. We validated our DR proposal running multiple hosts, communicating with each other over a network built on Docker swarm [40]. Our experimental setup considers four Virtual Machines (VMs) and a blockchain network consisting of the following components (see Fig. 3):

- A Certification Authority (CA Org1), on VM1;
- A Certification Authority (CA Org2), on VM2;
- An Orderer Service node, on VM1;
- A grid operator peer (Peer 1), on VM1;
- A market operator peer (Peer 2), on VM2;
- Customer peers (Peer 3 and 4), on VM3 and VM4;
- Clients on VMs for communication with peers.

The two organizations compose a consortium called DR Consortium (see Fig. 3) [41]. Members of an organization interact with the blockchain through smart contracts (chain codes) installed on peers. One dedicated smart contract, written in Go, takes care of the execution of the DR mechanism explained in section II. With this smart contract, the grid operator notifies the requests to increase or reduce the load in a given period of the day to avoid congestion on the network; the smart meter sends on the blockchain measurements of consumption; finally, customers receive DRtokens for their support in fulfilling the request by the grid operator.

On the day of the DR event, the customer queries the blockchain and receives the load target expected by the smart contract, calculated according to (6), then it participates in the DR event. At the end of the billing period, the market operator triggers the smart contract to check if customers' consumptions have been compliant with the grid operator's request and remunerates customers according to (9) assigning them energy-tokens, called DRtokens.

Consider two customers A e B, having baseline and measured load profile during the DR event as represented in Fig. 4, if the grid operator's request is to reduce the total load of 4 kW from 8:00 to 10:00 (DR1) and of 6 kW from 19:00 to 21:00 (DR2), the smart contract computes the desired load reduction for A and B. In order to split out the total load reduction, the smart contract needs the *Availability Profile* of customers, as requested by (6).

So, the first step is to compute the *Availability Profile* of each customer in order to find the most feasible for meeting the request by the network operator. The customers' initial *Availability* can be evaluated in two different ways:

1) Using (6), so for the first DR event the contribution will be proportional to the ratio (baseline of the customer)/(baseline of the aggregated loads), and for the subsequent DR events it will take into account the *Availability* evaluated following the first event using (4) and (5);
2) Performing training campaigns, simulating DR events, in order to collect data on customer response availability.

In this work, the authors considered the first method. A sample of customers has been simulated with related behavior. After some DR events, the result is that the *Availability Profile* of the customer A is higher than the *Availability Profile* of the customer B and assigns load reduction accordingly. Even if the baselines of both customers are similar, as can be seen from Fig. 4, during the two DR events the smart contract assigns to customer A a greater reduction compared to customer B (see the green lines in Fig. 4 to 5).

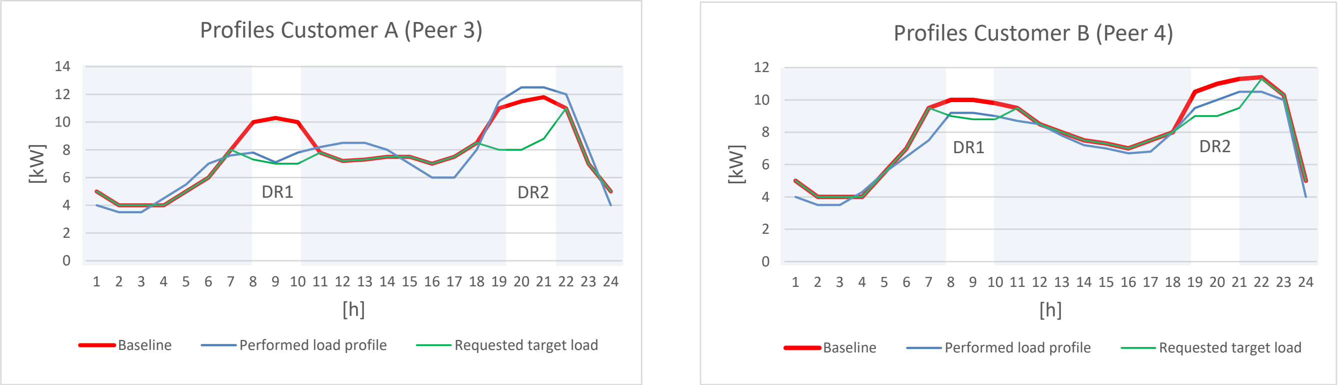
The baselines shown in Fig. 4 were assessed using the HighXofY averaging method with Y=10 and X=5. However, considering the use of second-generation smart meters [42], for this calculation, the authors considered a measure every 15 minutes, so the baselines are vectors of 96 elements instead of 24. In Figure 5, ΔP is the difference (represented if positive) between the baseline of the customer and the real consumption of the same customer and, as already said above, $\Delta P_{DR,h,i}$ is the capacity of the customer for a generic hour *h*, evaluated during the *i-th* DR events. From Figures 5 it is possible to understand that during DR1 both customers took part in the event and almost completely satisfied the request (the $\Delta P_{DR,h,i}$ of both is small), while during DR2 only customer B took part in the event. Consequently, for subsequent DR events, the *Availability* for those hours of customer A will be lower. Customers receive, through the blockchain, remuneration for the power reduction they have enforced.

At the end of the DR event, according to the difference between the baseline and the energy consumption recorded (ΔP performed by customer A and ΔP performed by customer B) the smart contract calculates the remunerations $r^{(A)}$ and $r^{(B)}$, namely the DR token amount for each hour according to equation (9), as described in listing 1. Figures **??** and 6 show the hourly token earnings for A and B, during the DR events hypothesized before, evaluated with (9).

These results were obtained simulating the network explained above and installing the smart contract on each peer that allows the implementation of a DR event. Figure 7 shows the response of the client of customer A after having invoked the function *getDRtoken* of the smart contract. In the same figure, the number of DRtokens for each hour and the total DRtokens earned by customer A are also represented.

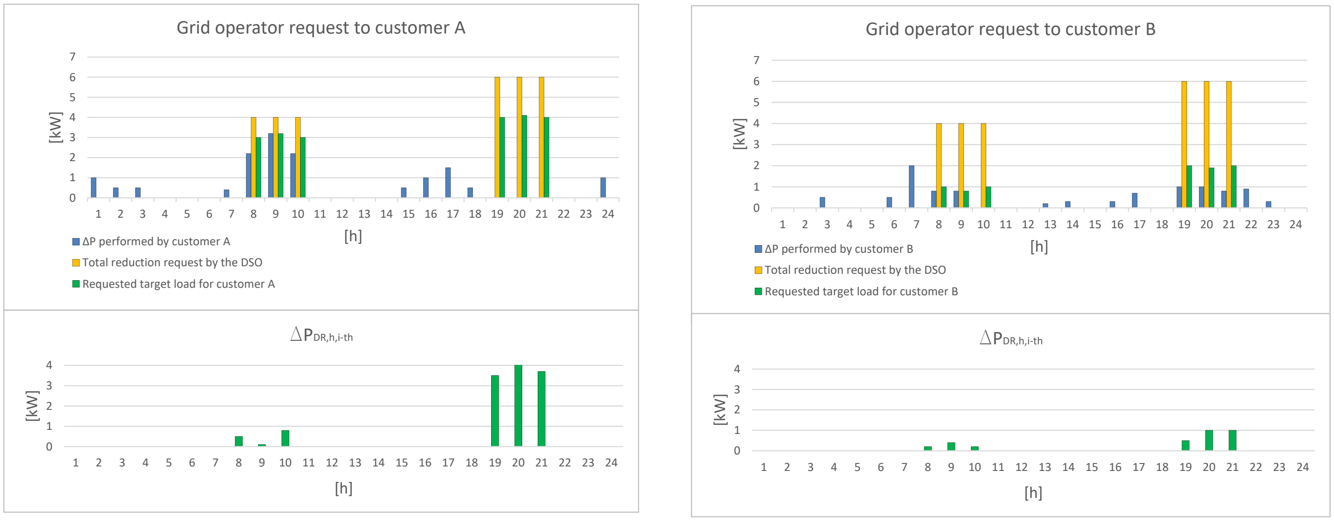
0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications



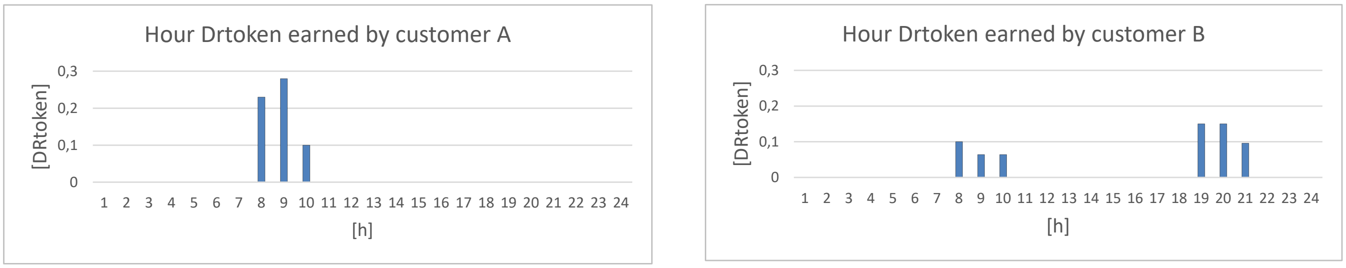
(a) (b)

Fig. 4: Baseline, performed load profile and requested target load customer A (a) and customer B (b). **(R2.7)**



(a) (b)

Fig. 5: Grid operator request, requested load reduction, ΔP and ΔP_{DR} for customer A (a) customer B (b) during DR1 and DR2. **(R2.7)**

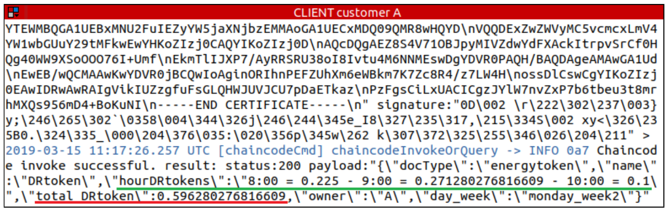


(a) (b)

Fig. 6: Hour DRtoken customer A (a) and B (b).

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications



```
YTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEMMAoGA1UECxMDQ09QMR8wHQYD\nVQQDExZwZWVyMC5vcmcxLmV4
YW1wbGUuY29tMFkwEwYHKoZIzj0CAQYIKoZIzj0D\nAQcDQgAEZ8S4V71OBJpyMIVZdwYdFXAckItrpvSrCf0H
Qg40WW9XSoOOO76I+Umf\nEkmTlIJXP7/AyRRSRU38oI8Ivtu4M6NNMEswDgYDVR0PAQH/BAQDAgeAMAwGA1Ud
\nEwEB/wQCMAAwKwYDVR0jBCQwIoAginORIhnPEFZUhXm6eWBkm7K7Zc8R4/z7LW4H\nossDlCswCgYIKoZIzj
0EAwIDRwAwRAIgVikIUZzgfuFsGLQHWJUVJCU7pDaETkaz\nPzFgsCiLxUACICgzJYlW7nvZxP7b6tbeu3t8mr
hMXQs956mD4+BoKuNI\n-----END CERTIFICATE-----\n" signature:"0D\002 \r\222\302\237\003}
y;\246\265\302`\0358\004\344\326j\246\244\345e_I8\327\235\317,\215\334S\002 xy<\326\23
5B0.\324\335_\000\204\376\035:\020\356p\345w\262 k\307\372\325\255\346\026\204\211" >
2019-03-15 11:17:26.257 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 0a7 Chaincod
e invoke successful. result: status:200 payload:"{\"docType\":\"energytoken\",\"name\"
:\"DRtoken\",\"hourDRtokens\":\"8:00 = 0.225 - 9:00 = 0.271280276816609 - 10:00 = 0.1\
",\"total_DRtoken\":0.596280276816609,\"owner\":\"A\",\"day_week\":\"monday_week2\"}"
```

Fig. 7: Request from Client node customer A.

Listing 1: Snippet of DR smart contract.

```
for k:=0; k<len(delta)-1; k++ {
    if delta[k]<=0 {
        delta[k]=0
        hourDRtoken=append(hourDRtoken,
        delta[k])
    } else if delta[k]<=deltaDesired[k] {
        r[k]=0.025*powerReduction[k] *
        (delta[k]*delta[k])
        hourDRtoken=append(hourDRtoken,
        r[k]*(hourwindow[k]))
    } else if delta[k]>deltaDesired[k] {
        delta[k]=deltaDesired[k]
        r[k]=0.025*powerReduction[k] *
        (delta[k]*delta[k])
        hourDRtoken=append(hourDRtoken,
        r[k]*(hourwindow[k]))
    }
    sum:=float64(0)
    for_,v:=range hourDRtoken {
        sum+=v
    }
    drtoken=sum
}
```

V. Conclusion

This work is an improvement of [43]. It shows the use of the blockchain technology and smart contracts for creating a reliable and transparent DR remuneration mechanism. In this way, an automatic system, in which prosumers can communicate with the grid operator to provide their flexibility, can be implemented. The blockchain technology chosen is the Hyperledger Fabric and it ensures that the same information is shared among all customers, with a sufficient degree of data protection and computational effort reduction. The grid operator notifies the request to increase/reduce the load in a given timeframe. The smart contract then computes the support that each customer can provide to fulfill the requested load modification and remunerates customers proportionally to their contribution with DRtokens. The improvements with respect to [43] consist in the CBL calculation, evaluated using a well known averaging method, and the evaluation of the customers' potential contribution to the next DR event, evaluated considering the "*Availability Profile*" of the customers. Moreover, a novelty was introduced, with the aim of improving data privacy in the network. Within the digital identity of each member of the network, a representative ID of the members was introduced, in order to determine permissions to use the different functions implemented by the smart contract and improve the privacy among the various network users. In this application, a new node has been added to the network (market operator) with the purpose to manage the identities of the customers. Results obtained in a virtual machine environment show the possibility to use blockchain technology to improve transparency and efficiency of the DR service in all market sessions, even considering privacy-preserving issues. Further work will be devoted to combining this application with others [44] to create a whole ecosystem of applications devoted to the electricity market.

References

[1] "Gartner Hypecycle for blockchain business 2019." [Online]. Available: https://www.ledgerinsights.com/gartner-blockchain-hype-cycle-2019/ accessed on october 7th 2019

[2] H. Lu, K. Huang, M. Azimi, and L. Guo, "Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks," *IEEE Access*, vol. 7, pp. 41 426–41 444, 2019.

[3] J. Wan, J. Li, M. Imran, and D. Li, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652–3660, 2019.

[4] O. Jogunola, A. Ikpehai, K. Anoh, B. Adebisi, M. Hammoudeh, S. Y. Son, and G. Harris, "State-of-the-art and prospects for peer-to-peer transaction-based energy system," *Energies*, vol. 10, no. 12, pp. 1–28, 2017.

[5] M. E. Peck, "Blockchains: How they work and why they'll change the world," *IEEE Spectrum*, vol. 54, no. 10, pp. 26–35, 2017.

[6] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, 2018.

[7] C. Park and T. Yong, "Comparative review and discussion on P2P electricity trading," *Energy Procedia*, vol. 128, pp. 3–9, 2017. [Online]. Available: https://doi.org/10.1016/j.egypro.2017.09.003

[8] J. Hwang, M. I. 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. [Online]. Available: https://doi.org/10.1016/j.egypro.2017.11.037

[9] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling Localized Peer-to-Peer Electricity Trading among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.

[10] X. Huang, C. Xu, P. Wang, and H. Liu, "LNSC: A Security Model for Electric Vehicle and Charging Pile Management Based on Blockchain Ecosystem," *IEEE Access*, vol. 6, pp. 13 565–13 574, 2018.

[11] M. L. Di Silvestre, P. Gallo, J. M. Guerrero, R. Musca, E. Riva Sanseverino, G. Sciumè, J. C. Vásquez, and G. Zizzo, "Blockchain for power systems: Current trends and future applications," *Renewable and Sustainable Energy Reviews*, vol. 119, no. September 2019, 2020.

[12] T. Sousa, T. Soares, P. Pinson, F. Moret, T. Baroche, and E. Sorin, "Peer-to-peer and community-based markets: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 104, no. October, pp. 367–378, 2019.

[13] J. Wu and N. K. Tran, "Application of Blockchain Technology in Sustainable Energy Systems : An Overview," *Sustainability 2018*, no. January, pp. 1–22, 2018.

[14] N. Wang, X. Zhou, X. Lu, Z. Guan, L. Wu, X. Du, and M. Guizani, "When energy trading meets blockchain in electrical power system: The state of the art," *Applied Sciences (Switzerland)*, vol. 9, no. 8, pp. 1–21, 2019.

[15] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, no. February 2018, pp. 143–174, 2019. [Online]. Available: https://doi.org/10.1016/j.rser.2018.10.014

[16] A. Ahl, M. Yarime, K. Tanaka, and D. Sagawa, "Review of blockchain-based distributed energy: Implications for institutional development," *Renewable and Sustainable Energy Reviews*, vol. 107, no. March, pp. 200–211, 2019. [Online]. Available: https://doi.org/10.1016/j.rser.2019.03.002

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2020.2992958, IEEE Transactions on Industry Applications

[17] C. Pop, T. Cioara, M. Antal, I. Anghel, I. Salomie, and M. Bertoncini, "Blockchain based decentralized management of demand response programs in smart energy grids," *Sensors (Switzerland)*, vol. 18, no. 1, 2018.

[18] European Union, "Regulation 2016/679," *Off. J. Eur. Communities*, vol. 2014, pp. pp. 1–88, 2016.

[19] A. C. Tsolakis, I. Moschos, K. Votis, D. Ioannidis, T. Dimitrios, P. Pandey, S. Katsikas, E. Kotsakis, and R. Garcia-Castro, "A Secured and Trusted Demand Response system based on Blockchain technologies," *2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA 2018*, 2018.

[20] N. Liu, X. Yu, C. Wang, C. Li, L. Ma, and J. Lei, "Energy-Sharing Model with Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3569–3583, 2017.

[21] D. Arnone, M. Mammina, S. Favuzza, M. Ippolito, E. Sanseverino, E. Telaretti, and G. Zizzo, "Demand project: Bottom-up aggregation of prosumers in distribution networks," pp. 1–6, 10 2018.

[22] C. Chen, J. Wang, S. Member, and S. Kishore, "for Large-Scale Residential Demand Response," *IEEE Trans. Power Systems*, vol. 29, no. 5, pp. 2219–2228, 2014.

[23] F. Wang, K. Li, C. Liu, Z. Mi, M. Shafie-Khah, and J. P. Catalao, "Synchronous pattern matching principle-based residential demand response baseline estimation: Mechanism analysis and approach description," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6972–6985, 2018.

[24] S. Mohajeryami, M. Doostan, A. Asadinejad, and P. Schwarz, "Error analysis of customer baseline load (CBL) calculation methods for residential customers," *IEEE Transactions on Industry Applications*, vol. 53, no. 1, pp. 5–14, 2017.

[25] conEdison, "Energy Efficiency and Demand Management Procedure – General Calculating Customer Baseline Load," CONSOLIDATED EDISON COMPANY OF NEW YORK, Tech. Rep., 2020.

[26] B. Xiang, K. Li, X. Ge, F. Wang, J. Lai, and P. Dehghanian, "Smart Households' Available Aggregated Capacity Day-ahead Forecast Model for Load Aggregators under Incentive-based Demand Response Program," *2019 IEEE Industry Applications Society Annual Meeting, IAS 2019*, pp. 1–10, 2019.

[27] T. K. Wijaya, M. Vasirani, and K. Aberer, "When bias matters: An economic assessment of demand response baselines for residential customers," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1755–1763, 2014.

[28] S. Mohajeryami, M. Doostan, and P. Schwarz, "The impact of Customer Baseline Load (CBL) calculation methods on Peak Time Rebate program offered to residential customers," *Electric Power Systems Research*, vol. 137, pp. 59–65, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.epsr.2016.03.050

[29] P. Faria, J. Spínola, and Z. Vale, "Methods for aggregation and remuneration of distributed energy resources," *Applied Sciences (Switzerland)*, vol. 8, no. 8, 2018.

[37] Hyperledger, "hyperledger-fabricdocs Documentation, Release master, chapter 4.12, Private Data," p. 441, 2019.

[30] D. C. Idoniboyeobu and S. L. Braide, "Analysis of Economic Load Dispatch, for Generation Expansion Planning," *International Journal of Scientific and Engineering Research*, vol. 8, no. 2, pp. 875–881, 2017.

[31] P. Faria, J. Spínola, and Z. Vale, "Aggregation and Remuneration of Electricity Consumers and Producers for the Definition of Demand-Response Programs," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 3, pp. 952–961, 2016.

[32] D. H. Vu, K. M. Muttaqi, A. P. Agalgaonkar, and A. Bouzerdoum, "Customer reward-based demand response program to improve demand elasticity and minimise financial risk during price spikes," *IET Generation, Transmission and Distribution*, vol. 12, no. 15, pp. 3764–3771, 2018.

[33] K. Srikanth Reddy, L. Panwar, B. Panigrahi, R. Kumar, and H. Yu, "Demand side management with consumer clusters in cyber-physical smart distribution system considering price-based and reward-based scheduling programs," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 2, pp. 75–83, 2017.

[34] P. Li, X. Guan, J. Wu, and D. Wang, "Pricing strategy for device-level demand response in a microgrid," *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, pp. 7579–7584, 2014.

[35] Hyperledger, "hyperledger-fabricdocs Documentation, Release master, chapter 4, Key Concepts ," p. 441, 2019.

[36] Hyperledger, "hyperledger-fabricdocs Documentation, Release master, chapter 4.5, Identity," p. 441, 2019.

[38] E. Androulaki, A. Barger, V. Bortnikov, S. Muralidharan, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Murthy, C. Ferris, G. Laventman, Y. Manevich, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *Proceedings of the 13th EuroSys Conference, EuroSys 2018*, vol. 2018-Janua, 2018.

[39] Hyperledger., "hyperledger-fabricdocs Documentation, Release master, chapter 1, Introduction," p. 441, 2019.

[40] H. C. Lee, "DCCN Docker Swarm Cluster Documentation," 2018.

[41] Hyperledger, "hyperledger-fabricdocs Documentation, Release master, chapter 7.3, Building Your First Network," p. 441, 2019.

[42] A. Pitì, G. Verticale, C. Rottondi, A. Capone, and L. Lo Schiavo, "The role of smart meters in enabling real-time energy services for households: The Italian case," *Energies*, vol. 10, no. 2, 2017.

[43] M. L. Di Silvestre, P. Gallo, E. R. Sanseverino, G. Sciume, and G. Zizzo, "A new architecture for Smart Contracts definition in Demand Response Programs," *Proceedings - 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe, EEEIC/I and CPS Europe 2019*, 2019.

[44] M. L. Di Silvestre, P. Gallo, M. G. Ippolito, E. R. Sanseverino, and G. Zizzo, "A technical approach to the energy blockchain in microgrids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4792–4803, 2018.

0093-9994 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Authorized licensed use limited to: Univ degli Studi di Palermo - Univ of Palermo. Downloaded on July 04,2020 at 11:55:00 UTC from IEEE Xplore. Restrictions apply.