Knirsch *et al. Energy Informatics* (2020) 3:2
https://doi.org/10.1186/s42162-020-0104-0

Energy Informatics

RESEARCH Open Access

Decentralized and permission-less green energy certificates with GECKO

Fabian Knirsch*, Clemens Brunner, Andreas Unterweger and Dominik Engel

*Correspondence:
fabian.knirsch@en-trust.at
Center for Secure Energy Informatics, Salzburg University of Applied Sciences, Puch b., Hallein, Austria

Abstract

A growing demand in sustainable energy harvested from renewable resources, such as wind or solar power, leads to new challenges in the electricity grid, which in future is also referred to as the *smart grid*. This also reflects in a more decentralized and diverse energy market. In such a market, prices do not only depend on production and demand, but also the source of energy production influences the price. In this paper, we present a decentralized and permission-less system for issuing, receiving and verifying *Green Energy Certificates for kWh Ownership* (GECKO) similar to the established *Renewable Energy Certificates* or *Green Tags*. These certificates allow to create a market for renewable energy. While the established system is designed for the wholesale market and does not allow for decentralized and permission-less verification, the proposed system is built on a blockchain-based approach and allows for the management and transfer of certificates. In our exemplary use case, Distribution System Operators (DSO) act as certification authority for privately or community owned power plants in regional energy markets. Customers can easily verify the integrity of such certificates without relying on a trusted third party or escrow service.

Keywords: Smart grid, Renewable energy, Blockchain, Distributed hash table

Introduction

Following national and international legislation (Morthorst 2003; European Comm 2017), the use of renewable energy and resources, such as wind and solar power, is of increasing interest for both, customers and utility companies. In particular, establishing and providing a market for tradable green energy certificates is one of the main objectives within the EU's strategy on implementing greenhouse gas reduction (Morthorst 2003). Furthermore, such a market is seen as one of the key components for the *smart grid*, a decentralized power grid based on mainly decentralized power plants and renewable resources (Monacchi and Elmenreich 2016). In order to establish such a smart grid, a large number of devices, wired and wireless meters has to be connected in an internet of things. Today, renewable energy is traded on both, a wholesale market between large utility companies and on a local neighborhood market, e.g., (Mengelkamp et al. 2018a, b). On the wholesale market in both, the US, with *Green Tags*, and the EU, with *Renewable Energy Certificates*, a system for such non-tangible certificates exists. However, these approaches are centralized and rely on selected certification authorities or escrow services. Furthermore, for customers it is impossible or infeasible to verify that energy is purchased from power plants that use renewable resources due to limited access to this wholesale market. Due to

these constraints, the existing scheme is not suitable for local energy markets where customers act as producers and consumers, given distributed energy resources such as small and privately or community owned wind, water or solar power plants.

In the directive 2009/28/EC on the promotion of the use of energy from renewable sources (The European Parliament and the Council of the European Union 2009), section 40 states that the certification process for renewable power plants must be "objective, transparent, non-discriminatory and proportionate".

In this work, we therefore address this gap by proposing a decentralized and permission-less scheme for energy certificates in local markets: GECKO, a system for *Green Energy Certificates for kWh Ownership*. The proposed scheme integrates into existing infrastructures of trading sites, but allows to certify power plants and produced kWh in a way that does not require a central trusted third party. This work does not assume physical links between producers and consumers. It rather regulates the demand for and supply of green energy via a market-based approach. Furthermore, consumers can easily verify whether purchased energy is produced from a certified power plant.

Contribution

To the best of our knowledge, GECKO is the first fully decentralized and permission-less system for issuing, receiving and verifying green energy certificates. The proposed approach allows verifiability by end-users and does not require trust in centralized parties. Furthermore, the proposed approach addresses concrete legal requirements (The European Parliament and the Council of the European Union 2009).

Our key contribution to achieve the aforementioned properties is twofold: (i) first, we set up a web of trust between certification authorities, which are the Distribution System Operators (DSOs); (ii) second, we show how these authorities certify power plants and how the certificates for kWh produced from these power plants can be traded and transferred, such that consumers can verify the integrity of the certificate. The approach builds on blockchain technology as a decentralized, permission-less and append-only database for storing events related to the production and consumption of kWh.

The proposed approach is evaluated with respect to scalability and security and a prototypical implementation is presented.

Structure

The rest of the paper is structured as follows: The "Related work" section discusses related work in the field and the "Preliminaries" section introduces the preliminaries of this work as well as the notation. The "GECKO" section presents GECKO in detail. Practical concerns regarding the implementation as well as security, scalability and complexity issues are discussed in the "Practical concerns" section before concluding the paper in the "Conclusion" section.

Related work

In Brunner (2017) and (2019), an approach for issuing and verifying educational certificates is presented. In this work, we extend this scheme and tailor it to the specific needs of local energy markets, especially focusing on regulatory and statutory requirements regarding the participants' roles and metering devices. Trading of energy in local markets is also proposed in Mengelkamp et al. (2018a, b), and similarly in Zhumabekuly Aitzhan

and Svetinovic (2016), Ilic et al. (2012) and Sikorski et al. (2017). These works present various forms of a blockchain-based approach for transactions within a local energy market focusing on trading and payment aspects. In contrast, in this work we focus on the transfer of green energy certificates and do not limit the application of blockchain technology to balancing supply and demand and the payment stream.

In Mihaylov et al. (2014a, b), the focus of the proposed scheme is on incentivizing customers to participate in load curtailment in a demand response setting. While this is realized via a blockchain-based approach, the source of the energy is not traced.

An overview of the economic challenges and benefits of integrating tradable green energy certificates in liberalized power markets is provided in Morthorst (2003). The work is motivated with the reduction of greenhouse gases within the EU. The authors conclude that energy certificates are a cost-effective way to achieve this goal, but provide no outlook to a technical implementation. In Hustveit et al. (2017), a similar analysis is conducted within the Swedish-Norwegian market with a stronger focus on the development of prices and the market.

For tradable green energy certificates, established systems exist in the US with *Green Tags*[1] and in the EU with *Renewable Energy Certificates* (Morthorst 2003). Such certificates include (among other data) a unique ID, information about the type of power plant, location and emission rate. In the US a *Renewable Energy Tracking System*[2] allows to follow the path from the certificate issuer to the consumer.

A summary of related work for tradable green energy certificates is shown in Table 1. We compare both, state of the art approaches presented for balancing energy supply and demand using blockchain technology and approaches for certification of green energy. Blockchain-based approaches aim to be customer-centered as they allow anyone to participate, whereas the well-established Green Tags (GT, as used in the US) and Renewable Energy Certificates (REC, as used in the EU) are more tailored towards the wholesale market. An approach is considered transparent if customers can easily track the origin and consumption of green energy. An approach ist trust-less if trust is spread over multiple instances. Private blockchains are not considered trust-less, as they often rely on single instances that maintain the permissions.

With GECKO, an approach is presented that builds on the concept of verifiable green energy in the form of non-tangible certificates, but specifically addresses local networks and neighborhood markets in a decentralized smart grid. A blockchain-based approach is chosen for a high level of transparency and decentralization, as well as for easy verifiability for customers.

Preliminaries

This section introduces the preliminaries by briefly introducing energy trading, defining actors, roles, terms and definitions and motivating the need for a decentralized and permission-less system.

Energy trading

Payment streams and physical energy flows are not necessarily linked together in energy trading. In contrast, energy is traded on wholesale or local markets mostly independent of

[1] Accessed Oct. 18, 2018. https://www.epa.gov/greenpower/renewable-energy-certificates-recs
[2] Accessed Oct. 18, 2018. https://www.epa.gov/greenpower/renewable-energy-tracking-systems

Table 1 Comparison of related work

	Scope	Transparent	Trust-less	Customer-centered
Mengelkamp et al. (2018b)	Balancing/payment	✓	–	✓
Mengelkamp et al. (2018a)	Balancing/payment	✓	–	✓
Zhumabekuly Aitzhan and Svetinovic (2016)	Balancing/payment	✓	–	✓
Ilic et al. (2012)	Balancing/payment	–	–	✓
Sikorski et al. (2017)	Balancing/payment	–	–	✓
GT (US)	Certification	–	–	–
REC (EU)	Certification	–	–	–
GECKO	Certification	✓	✓	✓

GT refers to Green Tags, as used in the US and REC refers to Renewable Energy Certificates as used in the EU.

existing physical infrastructures and actual energy flows. Following this principle, purely market-based approaches and approaches that assume such a separation have been shown in e.g., Mashhour and Moghaddas-Tafreshi (2010); Ramchurn et al. (2011); Monacchi and Elmenreich (2016); Mengelkamp et al. (2018a) and the concept is similarly reflect in statutory frameworks, such as The European Parliament and the Council of the European Union (2009).

Similarly, this paper does not address the physical aspects of energy transportation and energy flows, but rather focuses on the market perspective and a verifiability of power plants and purchased energy, which is represented by the consumption of green energy certificates. The term *purchase* is used for kWh that are bought on the market, whereas the term *consume* is used for certificates for kWh that are used and therefore not longer valid. Since this paper focuses on the lifecycle of certificates, the terms are used synonymously, i.e., purchased certificates are consumed immediately. In addition, while producers and consumers of kWh are represented by smart meters that produce and consume actual energy, the certificates for these units of energy are traded independently and as such, green energy can be purchased on the market without the need of a physical link.

Actors and roles

For the proposed use case the following actors and roles are defined:

- **DSO:** The DSO is an authority that issues certificates to power plants harvesting energy from renewable resources and to smart meters purchasing renewable energy. In practice, the local DSOs act as such certifiers. They establish a confirmation network related to a web of trust and therefore also certify each other. DSOs closely interact with small and community owned power plants within their sphere and – in some regions (European Commission 2012) – also provide the metering infrastructure. As will be shown later, power plants may obtain certificates from more than one certifier to strengthen their asset.
- **Producer:** The producer represents a power plant, which is a privately or community owned facility that produces energy from renewable resources such as water, wind or solar power. Such a power plant wants to obtain a certificate, such that each produced kWh can be signed and are therefore certified as well. These certified kWh can then be traded by transferring the certificate from the seller to the

buyer. A certified smart meter measures the amount of energy fed into the system and issues a new certificate for each kWh.

- **Consumer:** Consumers are interested in buying certified green energy from either an energy provider or on a peer-to-peer local energy market. Consumers can easily verify the integrity of the certificate with the proposed scheme. A certified smart meter in the customer premises purchases certified kWh and marks consumed energy certificates as used.

Figure 1 shows an overview of the actors involved in GECKO, their roles and how they interact. All data is stored on a blockchain in chronological order. The process of publishing a state change (e.g., creating or consuming a certificate for green energy) is referred to as an *event*. First, DSOs certify both, the producers' power plants and the consumers' smart meters. These certificates with the digital signature of the DSO are written to the blockchain, so that everyone can verify their integrity and validity. For each kWh generated from such a certified power plant, a producer creates a new certificate and writes an event to the blockchain. The so created green energy certificate is now traded to a consumer. The consumer of the corresponding kWh can check the origin and also the publicly available information of the DSO that certified the issuing smart meter. Once the corresponding kWh is used by the consumer, this is again written to the blockchain and the green energy certificate is marked as used. The costs incurred by this process for the issuer and the user are discussed in "Practical concerns" section.

Terms and notation

Throughout this paper the notation as shown in Table 2 is used. DSOs are denoted as D_i with an index. The producers and consumers of DSO D_i are denoted as P_i^j and C_i^k, respectively. The set of all DSOs is denoted as $\mathbb{D}$ and the set of producers and consumers of a specific DSO D_i are denoted as $\mathbb{P}_i$ and $\mathbb{C}_i$, respectively. In addition, the unified set of all producers, consumers and DSOs is referred to as

$$\mathbb{A} = \mathbb{D} \cup \mathbb{P} \cup \mathbb{C},$$

where

$$\mathbb{D} = \bigcup_i \mathbb{D}_i \text{ and } \mathbb{P} = \bigcup_i \mathbb{P}_i \text{ and } \mathbb{C} = \bigcup_i \mathbb{C}_i.$$

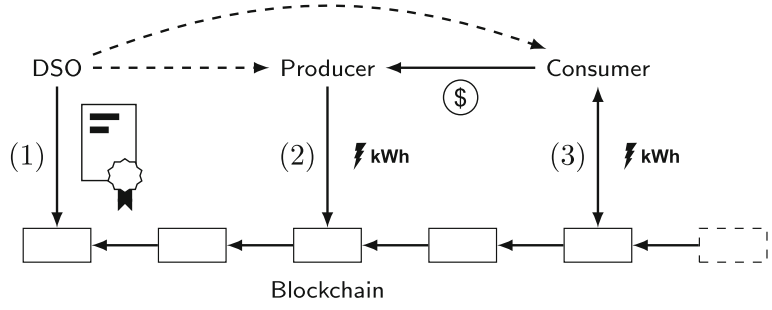

Fig. 1 In GECKO, DSOs issue certificates for power plants (1) and producers use these certificates to sign produced units of renewable energy (2). This information is stored on a blockchain chronologically ordered and consumers can verify the integrity and origin of certificates (3). Solid arrows from DSO, producer and consumer to the blockchain denote data that is written to or read from blocks and dashed arrows from DSO to producer and consumer denote the certificate information contained in that data

Table 2 Notation used in this paper

Symbol	Meaning
D_i	DSO with index i
P_i^j	Producer j of D_i
C_i^k	Consumer k of D_i
$\mathbb{D}$	Set of DSOs
$\mathbb{P}_i$	Set of producers of D_i
$\mathbb{C}_i$	Set of consumers of D_i
$\mathbb{P}$	Set of all producers
$\mathbb{C}$	Set of all consumers
$\mathbb{A}$	Set of all DSOs, producers and consumers
$\mathbb{E}$	Set of all events
$\mathrm{E}_X(p_1, p_2, \ldots, p_M)$	Event of type X with parameters $p_{1\ldots M}$
$G_{i,j}^l$	Green energy certificate of P_i^j with index l

Further, dependent events are denoted as $[\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_N | R] \leftarrow$ dependent, where event dependent requires that events $\mathrm{E}_{1,\ldots,N}$ occurred before and an optional requirement R is met.

An event $\mathrm{E}(p_1, p_2, \ldots, p_M)$ with $p_i, i = 1, \ldots, M$ parameters describes a unique and distinguishable state change of an actor or energy certificate that occurred at some point in time.

A green energy certificate $G_{i,j}^l$ is a digital document that is created through an event by producer P_i^j, confirmed by the DSO D_i. Each certificate has a sequence number l – representing the l-th kWh –, an issuing party and a timestamp. The scope of the sequence number is the issuer.

Blockchain

Blockchain technology was originally proposed in 2008 by Nakamoto (2008) for financial transactions and has since then been applied to many fields, e.g., for the Internet of Things (Christidis and Devetsikiotis 2016; Han et al. 2018) or in the energy domain (Knirsch et al. 2018; Mengelkamp et al. 2018b; Munsing et al. 2017). With Ethereum Wood (2017), a generic blockchain that even allows the execution of programs has been proposed. Ethereum will be used in this paper for our prototypical implementation.

A public blockchain allows decentralized, trust-less and append-only data storage. A number of nodes establish a peer-to-peer network and store a common state. For adding a new data item d, it is sent in the form of transactions. One or more of these transactions become part of a new block B_i that is immutably appended to the previous block B_{i-1} by using a cryptographic hash function $\mathrm{H}(\cdot)$. A block therefore is represented by $B_i = (\mathrm{H}(B_{i-1}), d)$.

For sending a transaction, a public/private key pair is created and the transaction is signed with the private key of the sender. All participants store a local copy of the data and agree on the current common state. For selecting the participant that is allowed to append a new block, a number of algorithms have been proposed, e.g., proof of work, proof of stake and proof of authority. In some cases, e.g., for (financial) transactions in Bitcoin (Nakamoto 2008) and for implemented smart contracts in Ethereum Wood (2017), the consensus algorithm is built into each node, i.e., new blocks are only accepted if all transactions in this block yield a valid state change.

A blockchain, however, can also be used as a decentralized, append-only database, where nodes agree on the chronological order of data written to the blockchain without checking the state and consistency of that data itself. Bitcoin, for instance, is also capable of embedding a limited amount (e.g., 80 bytes in OP_RETURN) (Bartoletti and Pompianu 2017) of additional data in transactions that is not checked by the consensus algorithm.

Distributed file systems

If the consensus algorithm is not built into the blockchain itself, the data needs to be processed on the application layer. In GECKO, events are data stored on the blockchain. This data is not incorporated in the consensus of the blockchain, but rather is checked off-chain by the applications on the producing and consuming devices. The data represents an event, which is the hash of a file that is stored in a *distributed file system* or a distributed hash table (Benet 2014). Multiple events from one sender can be combined into a list and stored in one file. The hash of this file is then locked in a public blockchain. Applications receive events by reading the chronologically ordered hashes from the blockchain and loading the corresponding data from the distributed file system. It can then be checked if the order of the events is valid according to the GECKO protocol as formally described in the next section.

GECKO

In this section, we describe GECKO, a decentralized system for issuing, receiving and verifying Green Energy Certificates for kWh Ownership. Green energy certificates are produced by a smart meter, which is certified by the local DSO. These certificates can then be traded and consumed by another certified smart meter. The production, transfer and consumption of green energy certificates is represented as a global state of ordered events. This global state is maintained in a blockchain and thus allows all participants to view and verify the currently produced and hence consumable certificates. GECKO therefore does not require a centralized trusted party for handling the green energy certificates, but instead trust is spread over all participants.

A valid consumable green energy certificate is defined by a series of events, including a set of published accounts, certified producers and consumers and the actual production and consumption of the certificate. For writing events to the blockchain, a public/private key pair is used. The public key represents the organization (e.g., a DSO or a consuming and producing smart meter) and the private key is needed to write data to the blockchain and add events to GECKO, respectively.

Creating and publishing accounts

For being publicly verifiable, each DSO requires a public account. For establishing such an account, DSO D_i needs to trigger the $\mathrm{E}_{\mathrm{Publish}}$ event without prerequisite events (denoted as $\cdot$), but its public key must not have been used for an actor in GECKO before:

$$[\,\cdot | D_i \notin \mathbb{A}] \leftarrow \mathrm{E}_{\mathrm{Publish}}(D_i).$$

This event writes all necessary data to identify the DSO on the blockchain. This includes name, address, website and contact information. In addition, the DSO may back its profile by adding traditional means of authentication such as X.509 certificates.

Private accounts need to create a public/private key pair as well, but can skip the publish event and thus stay pseudonymous. This is similar to public blockchains such as the Bitcoin (Nakamoto 2008) or Ethereum (Wood 2017) network. Note that this allows for customers to remain pseudonymous within the blockchain and having their identity only known to the local DSO. The latter is acceptable, as in practice there exists a tight business relation between the customer and the DSO for receiving energy from the grid and feeding energy to the grid. A detailed discussion of privacy aspects within blockchain-based certificates is conducted in Brunner et al. (2019)

Certifying green energy producers and consumers

Green energy certificate producers and consumers need to be certified by their DSO. To do so, they contact the local DSO using the contact information from the previous $\mathrm{E}_{\mathrm{Publish}}$ event. In case of a producer, the DSO inspects the green energy power plant for the fulfillment of the requirements for sustainable energy production. A trusted hardware device, e.g., a smart meter, ensures that produced kWh cannot be manipulated. In case of a consumer, only a trusted hardware device is required for correctly recording the purchased kWh.

Before triggering the $\mathrm{E}_{\mathrm{Certify}}$ event, the DSO has to make sure that a customer (producer or consumer) has not been certified yet by another DSO as a producer or consumer and is not another DSO itself. As described above, the DSO has to have published its account and contact information:

$$\begin{aligned}\left[\mathrm{E}_{\mathrm{Publish}}(D_i)|P_i^j \notin \mathbb{A}\right] &\leftarrow \mathrm{E}_{\mathrm{Certify}}\left(P_i^j\right),\\ \left[\mathrm{E}_{\mathrm{Publish}}(D_i)|C_i^k \notin \mathbb{A}\right] &\leftarrow \mathrm{E}_{\mathrm{Certify}}\left(C_i^k\right).\end{aligned}$$

Producing green energy certificates

In order to produce a green energy certificate, at least one certified producer is needed, e.g., producer P_i^j of DSO D_i. For each kWh produced from a sustainable energy resource, a unique green energy certificate $G_{i,j}^l$ is generated, triggering an $\mathrm{E}_{\mathrm{Produce}}$ event:

$$\left[\mathrm{E}_{\mathrm{Certify}}\left(P_i^j\right)\right] \leftarrow \mathrm{E}_{\mathrm{Produce}}\left(P_i^j, G_{i,j}^l\right).$$

Consuming green energy certificates

For consuming a green energy certificate it needs to be transferred to the consumer, e.g., through trading in local energy markets. The owner of the green energy certificate, i.e., the producer, triggers an $\mathrm{E}_{\mathrm{Consume}}$ event as soon as the certificate is sold or transferred.

For this to be possible, the certificate $G_{i,j}^l$ for the corresponding kWh has to have been produced, but never consumed before, and the consumer has to have been certified by the same local DSO D_i:

$$\begin{aligned}&\left[\mathrm{E}_{\mathrm{Certify}}\left(C_i^k\right), \mathrm{E}_{\mathrm{Produce}}\left(P_i^j, G_{i,j}^l\right) \mid\right.\\ &\left.\nexists C \in \mathbb{C} : \mathrm{E}_{\mathrm{Consume}}\left(C, G_{i,j}^l\right)\right] \leftarrow \mathrm{E}_{\mathrm{Consume}}\left(C_i^k, G_{i,j}^l\right).\end{aligned}$$

This restriction of trading only in a local energy market (i.e., all producers and consumers are certified by the same DSO) can be lifted by allowing DSOs to establish a confirmed network of DSOs, as explained in the next section.

Confirmed DSO network

We assume that customers are able to verify their own DSO, based on the $\mathrm{E}_{\mathrm{Publish}}$ event. To create a decentralized trustworthy network of DSOs we introduce a confirmation network. Each DSO can confirm other DSOs by triggering the $\mathrm{E}_{\mathrm{Confirm}}$ event. In order to successfully trigger such an event, both DSOs must have published an account before and the sets of producers and consumers $\mathbb{P}_i \cup \mathbb{C}_i$ and $\mathbb{P}_j \cup \mathbb{C}_j$, respectively, must not overlap. Formally, this is represented as:

$$\begin{aligned}&\left[\mathrm{E}_{\mathrm{Publish}}(D_i), \mathrm{E}_{\mathrm{Publish}}(D_j) \right| \\ &\left.(\mathbb{P}_i \cup \mathbb{C}_i) \cap (\mathbb{P}_j \cup \mathbb{C}_j) = \emptyset\right] \leftarrow \mathrm{E}_{\mathrm{Confirm}}(D_i, D_j).\end{aligned}$$

If there exist two $\mathrm{E}_{\mathrm{Confirm}}$ events, $\mathrm{E}_{\mathrm{Confirm}}(D_i, D_j)$ and $\mathrm{E}_{\mathrm{Confirm}}(D_j, D_i)$, this means that the corresponding DSOs are connected. If DSOs are connected, all producers and consumers are also connected, until one of the events is revoked.

Similar to the $\mathrm{E}_{\mathrm{Consume}}$ event for a single DSO, the kWh has to have been produced, but never consumed before. In the case of connected DSOs D_m and D_i, the producer P_m^n and the consumer C_i^k may have been certified by different DSOs. This is formally expressed as:

$$\begin{aligned}&\left[\mathrm{E}_{\mathrm{Certify}}\left(C_i^k\right), \mathrm{E}_{\mathrm{Produce}}(P_m^n), \right. \\ &\mathrm{E}_{\mathrm{Confirm}}(D_i, D_m), \mathrm{E}_{\mathrm{Confirm}}(D_m, D_i) \mid \\ &\left.\nexists C \in \mathbb{C} : \mathrm{E}_{\mathrm{Consume}}(C, G_{i,j}^l)\right] \leftarrow \mathrm{E}_{\mathrm{Consume}}\left(C_i^k, G_{m,n}^l\right).\end{aligned}$$

Note that confirmations are neither commutative, i.e,

$$\mathrm{E}_{\mathrm{Confirm}}(D_i, D_j) \not\Rightarrow \mathrm{E}_{\mathrm{Confirm}}(D_j, D_i),$$

nor transitive, i.e.,

$$\mathrm{E}_{\mathrm{Confirm}}(D_i, D_j) \wedge \mathrm{E}_{\mathrm{Confirm}}(D_j, D_k) \not\Rightarrow \mathrm{E}_{\mathrm{Confirm}}(D_i, D_k).$$

In a decentralized network this assures that no malicious DSO is able to establish a connection without mutual confirmation. For producers and consumers the local DSO is a trusted entry point into the network of confirmed DSOs.

Practical concerns

This section discusses practical concerns for implementing the proposed GECKO protocol. First, its is described how consensus is achieved and how nodes check events. Second, details of the implementation are presented and finally, scalability and security issues are discussed.

Achieving consensus

The consensus algorithm of the GECKO protocol is decoupled from the consensus algorithm of the blockchain. In fact, the proposed protocol is not limited to a specific blockchain or type of consensus and thus can be tailored to specific needs of different fields of applications in different markets, e.g., public blockchains or private blockchains.

In GECKO, a smart meter or DSO $a \in \mathbb{A}$ creates events. Event data from one or more events $\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_N$ are stored in a distributed hash table and the hash reference $\mathrm{H}(\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_N)$ is locked in the blockchain. This is depicted in Fig. 2. A hash may refer

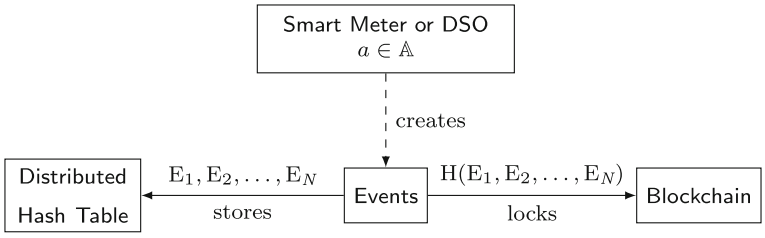

Fig. 2 A smart meter or DSO $a \in \mathbb{A}$ creates a series of one or more events $E_1, E_{2,\ldots,}E_N$ and stores the event data in a distributed hash table. The corresponding hash is locked in the blockchain. In order to retrieve a list of events, the hashes are read from the blockchain and the hash is used as the key to load event data from the storage

to more than one event and therefore allows for easy scalability. The blockchain is only used as a read-only, decentralized and permission-less data storage with an immutable chronological ordering of events and hashes, respectively.

Each node, i.e., in particular each certified smart meter, is responsible for checking the event data before processing and accepting events as valid. It therefore reads hashes stored in newly created blocks, uses these hashes to retrieve the corresponding raw data from the distributed hash table and checks whether the event data follows the rules presented in the previous section. Valid events add to an internal update of the state while invalid events are ignored.

Scalability

For applications building on blockchain technology, scalability and throughput are major concerns (Croman et al. 2016). Scalability can be measured in transactions per second.

GECKO supports multiple levels of scalability. Table 3 summarizes the scalability features of GECKO and the implications in terms of worst-case delay of the various types. Scalability can be improved by not creating a transaction for each kWh certificate individually, but by allowing multiple kWh certificates in a single transaction. As described in the previous section, the protocol allows multiple E_{Produce} events to be sent within a single transaction. If, for instance, all produced kWh for one day are sent only once, this reduces the total number of transactions per year and per household to 365, which reduces the number of transactions by 98.5%. This comes, however, at the cost of being less flexible in terms of real-time markets.

Each transaction carries a hash reference of data stored in a DHT and a signature. To further improve the scalability, the smart contract is adopted to allow registering multiple hash reference and signature pairs within a single blockchain transaction. This is referred

Table 3 Types of scalability in GECKO

Type	Data per transaction	Use of proxy	Worst-case delay
Naïve	1 event		
Multiple events	n events		$n - 1$ events
Proxy transaction	k hash references/signatures with 1 event each	Yes	$k - 1$ events
Proxy transaction with multiple events	k hash references/signatures with n events each	Yes	$(n - 1) \cdot (k - 1)$ events

as to as proxy transaction. In the proposed setup, DSOs can create such proxy transaction for their clients.

Security analysis

In the proposed protocol three roles exist: DSOs, producers and consumers. This section performs a security analysis for each of the roles in the face of a malicious attacker. Malicious attackers can alter and reroute messages, break communication links and change the protocol.

In decentralized trustworthy networks, clusters of malicious actors can form and then undermine the validity of confirmations. In the context of this paper a group of malicious DSOs could certify producers of non-renewable energy (e.g., nuclear power plants) and confirm one another in order to propagate their non-renewable producers to other local networks. This is also referred to as a Sybil attack (Douceur 2002). Figure 3 depicts one possible form of a Sybil attack where a group of malicious DSOs (black nodes) are connected (dashed line) via a single link to a group of honest interconnected DSOs (white nodes).

Detecting such Sybil attacks has been studied extensively in literature, see e.g., Danezis and Mittal (2009); Yu et al. (2008); Conti et al. (2012). Most commonly, detection approaches require formalizing the network as a graph.

The network of mutually confirmed DSOs can be represented by an undirected graph $G = (\mathbb{V}, \mathbb{L})$. The set of vertices $\mathbb{V} := \mathbb{D}$ is the set of all DSOs and the set of edges or lines

$$\mathbb{L} := \{(D_i, D_j) : \exists \mathrm{E}_{\mathrm{Confirm}}(D_i, D_j) \wedge \exists \mathrm{E}_{\mathrm{Confirm}}(D_j, D_i), i \neq j\}$$

results from all DSOs which have mutually confirmed each other, as described in "GECKO" section via $\mathrm{E}_{\mathrm{Confirm}}$ events.

G is undirected, because the $\mathrm{E}_{\mathrm{Consume}}$ event requires mutual confirmations from both DSOs involved, i.e., the DSO that certified the producer and the DSO that certified the consumer. For the analysis of Sybil attacks in the graph, only mutually confirmed and thus connected relationships are considered.

As described above, Sybil attacks can be detected and mitigated for undirected graphs. When consuming energy from a producer outside the local network, implementing one of the mitigation approaches is reasonable as long as the additional thoroughness outweighs the additionally consumed energy.

Generally, producers can act maliciously and flood the network with fake certificates, i.e., certificates for energy from non-renewable resources or certificates for energy that

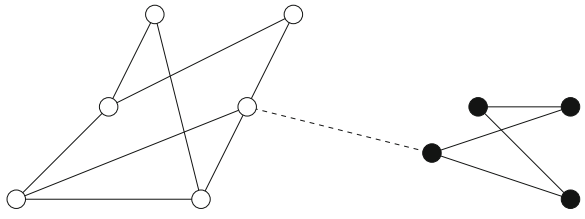

Fig. 3 Visual representation of a Sybil attack where a group of malicious DSOs (black nodes) are connected (dashed line) via a single link to a group of honest DSOs (white nodes)

has not actually been produced. In order to prevent this misbehavior, DSOs have to regularly audit the tamper-proofness of the smart meters and whether producers comply to the requirements for green energy. The tamper-prrofness of smart meter is regulated in, e.g., Technische Richtlinie BSI TR-03109 (2015). If DSOs fail to detect such malicious behavior their reputation may decrease in the web of trust by losing confirmations from other DSOs. Similarly, consumers can act maliciously by attempting to consume certificates for green energy multiple times. Again, this can be prevented by having a regular audit for the tamper-proofness of the smart meters by the local DSO.

Prototypical implementation

This section describes the prototypical implementation of GECKO and briefly outlines the technologies used. This implementation also serves as the basis for the complexity analysis.

Since the consensus algorithm of the GECKO protocol is decoupled from the consensus algorithm of the blockchain, the architecture is split into three main modules.

In a practical setup, GECKO either runs directly on a smart meter or sealed hardware, which is certified by the local DSO, or the smart meters use the API to call a GECKO on a trusted device. Similarly, if a third party wants to validate the correctness of issued green energy certificates there are two options (i) using a global instance of GECKO and trusting the hosting party; or (ii) installing a GECKO node locally or on a trusted device and using it to verify the correctness of issued green energy certificates.

The modules and the deployment is illustrated in Fig. 4. The following modules are deployed on the smart meter or a sealed hardware:

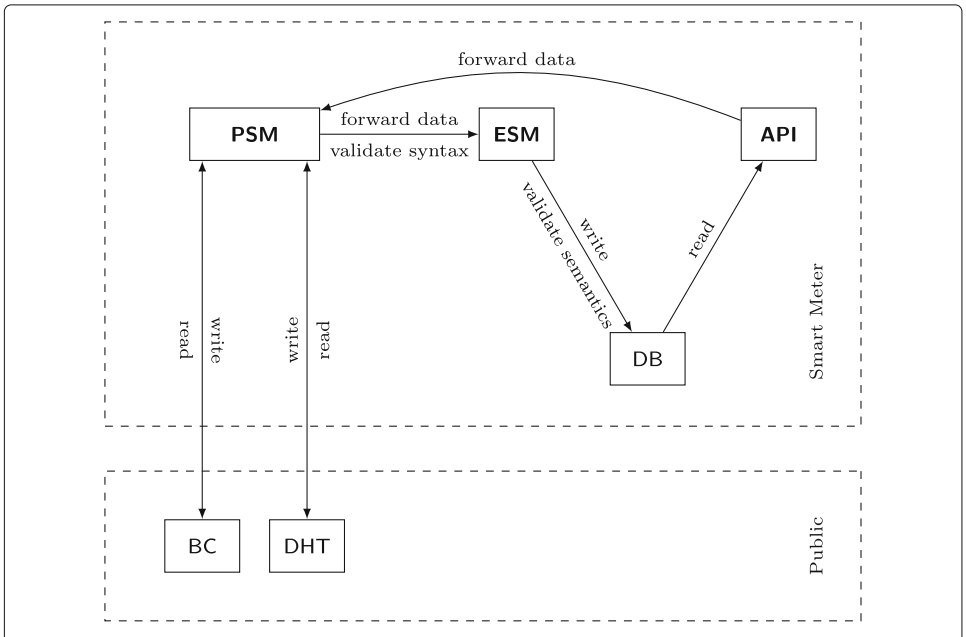

Fig. 4 The prototypical implementation consists of three main modules: The PSM communicates with the Blockchain (BC) and a distributed hash table (DHT). Syntactically correct data is forwarded to the ESM, which validates the semantics, updates the current internal state and stores the new state in a local database (DB). The API allows to read data from the DB and to send data to the PSM for publishing new events to the BC and the DFS, respectively. While BC and DHT store data publicly, the rest of the modules is deployed on the smart meter

- **Public Storage Module (PSM):** The PSM handles the reading and writing of data from and to the blockchain and from and to the distributed file system (also referred to as a distributed hash table). IPFS (Benet 2014) is used for the prototypical implementation. All data is processed on the application layer and the GECKO protocol is not incorporated in the blockchain itself and thus not limited to a particular blockchain implementation. For the prototypical implementation, an Ethereum smart contract is used that triggers the PSM if a new GECKO event has been added to the blockchain. In the blockchain, this event is locked as a hash and the PSM reads the corresponding file from the distributed hash table. The content of the files is syntactically validated with respect to the required fields (e.g., publish events must contain name, address, website and contact information) and valid events are forwarded to the Event State Machine module for processing.
- **Event State Machine (ESM):** The ESM validates the semantics and maintains the inner state of events for a GECKO node. Incoming events lead to a state change in the GECKO protocol if and only if all required dependent events (as described in "GECKO" section) have been executed before. Since each transaction in the blockchain has a timestamp, this chronological order is immutable and easily verifiable. Invalid events or events that would lead to invalid states are ignored. In order to persist the state of the ESM (i.e., the current state of published DSO accounts, valid and consumed green energy certificates, etc.), a local database is used. This allows for fast and easy access through the Application Programming Interface.
- **Application Programming Interface (API):** The API is provided for external devices to communicate with the GECKO protocol. While GECKO is designed to run on a smart meter, in some cases it might be desirable to access a GECKO node through the API, e.g., for small or lightweight devices that do not have the computational power required to run a GECKO instance. Read requests to the API only communicate with the local database and write requests, which in turn create a new event in the GECKO protocol, then create a file in the distributed file system and forward the hash of that file as a signed transaction to the Ethereum smart contract.

In Wüst and Gervais (2018), the criteria for using a public blockchain such as Ethereum are as follows: need for storing a state with multiple writers and no trusted third party, where writers are not known and not trusted. Since GECKO is a public and permissionless platform that does not limit the users, nor does it rely on a single trusted third party, a public blockchain qualifies for the permanent record of the hashes.

The smart contract that is used for the prototypical implementation is publicly available and usable[3]. The costs for the deployment on March. 17, 2019[4] were about 0.01 Ether (approx. US$ 1.44) and the costs for writing a single event to the blockchain were about 0.00058 Ether (approx. US$ 0.08). While an amount of eight cents seems a lot per kWh at first, it has to be noted that for improved scalability multiple kWh can be traded within a single transaction. However, this can be further reduced with proxy transactions as described previously. Figure 5 shows the data points for the costs for registering 1, 5, 10, 50, and 100 hash references and signature pairs within a single transaction and a solid fitting curve. With the use of a proxy transaction it is possible to reduce the costs per actor

[3] Smart contract address (Kovan testnet): 0x7fe8dD765D696bf9391AAad6D03025C823DBB566
[4] 1 Ether = US$ 139, avg. Gas price = $1.457 \cdot 10^{-8}$ Ether, Source: etherscan.io

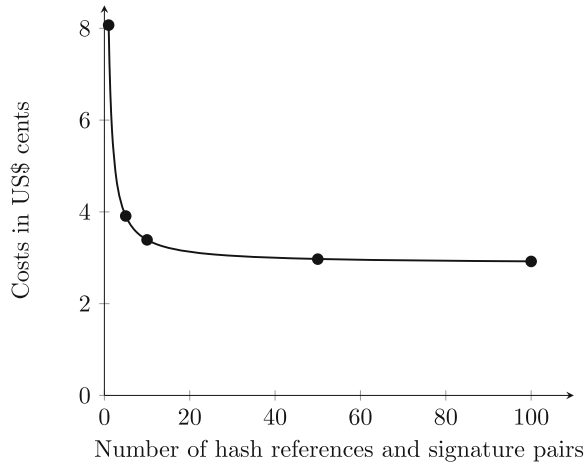

Fig. 5 This figure shows the costs for registering multiple hash references and signature pairs withing a single transaction for numbers of 1 to 100 pairs

to 0.0003 Ether (approx. US$ 0.029) by registering multiple events. This is 31.41% cheaper than a standard transaction[5].

The prototypical implementation has shown that GECKO can be used on a public blockchain at reasonable costs. For practical applications, the scalability is only limited by the real-time requirements of the application. If a GECKO event is required to be instantly written to the blockchain (e.g., once a full kWh has been generated), this will require a single transaction for each event. By trading multiple kWh (and thus multiple events) in a single transaction or by using proxy transactions the costs significantly decrease, but the time until a kWh can be traded increases.

For the cost analysis the solar photo voltaic electricity production in Germany is used as a reference. Germany has the largest number of installed photo voltaic power plants in Europe (Wirth 2018) and thus is a representative market for cost and scalability assessments. As of 2016[6], 1,623,467 photovoltaic power plants are installed in Germany. This yields a net production of 38,098,000,000 kWh that is fed into the public electricity grid. On average, approx. 23,467 kWh are produced per household. Each smart meter therefore would trigger approx. 23,467 $E_{Produce}$ events within one year, which would be on average 64 $E_{Produce}$ events per day and per producer.

Given the 64 $E_{Produce}$ events per day and producer, this implies costs of US$ $0.08 \cdot 64 = 5.12$ in total per actor. However, given the level of scalability with proxy transactions containing multiple events this drops the costs per actor to US$ 0.029 for up to 100 $E_{Produce}$ events issued simultaneously. This implies costs of US$ $0.029 \cdot 64 = 1.856$ in total per day and producer.

Complexity

In this section, the worst-case time complexity of the operations in the ESM is analyzed. The ESM maintains the inner state of a GECKO node and therefore the validity of green energy certificates. An incoming event triggers a state change if all dependent events have been executed before. Given the protocol definition in "GECKO" section, all events are

[5]The cost for a transaction without additional data on Ethereum is 21000 Gas, which is approx. US$ 0.04
[6]Accessed Oct. 18, 2018. https://www.foederal-erneuerbar.de/

analyzed with a naïve approach, i.e., without optimizations in the algorithm, and with an optimized approach, i.e., with the theoretically optimal algorithm known.

Table 4 shows a summary of the complexity analysis of operations in the ESM when executing the GECKO protocol and Table 5 extends this for confirmed DSO networks.

For the naïve approach, the complexity of the $E_{Publish}$ event depends on the number of elements in the set of all DSOs, producers and consumers, since it needs to be checked by the ESM if such an event has been published before. Similarly, the $E_{Certify}$ event depends on the number of elements in this set and also on the number of elements in the set of all (already published) events. For the $E_{Produce}$ and the $E_{Consume}$ events, the ESM needs to check whether they already occurred in the set of all previous events. With an optimized algorithm (e.g., a hash table Cormen et al. (2001)), all this operations can be performed in constant time, which allows for an implementation of the ESM in an environment with limited computational resources.

When extending the GECKO protocol with confirmed DSO networks, the complexity of the $E_{Confirm}$ event increases for the naïve approach due to the need for set intersection. For the optimized approach, a Bloom filter can be used instead, which reduces the time complexity to constant time (Cormen et al. 2001).

Real-time considerations

This section discusses the requirements for the blockchain implementation used as a basis for GECKO. Since a cryptographic hash reference to the data is stored on-chain, this only requires the blockchain of being capable of handling between 256 bit and 512 bit of additional data (depending on the cryptographic hash function) (Menezes et al. 2001).

In Knirsch et al. (2018), it is shown that Bitcoin would suffice for storing the required hash reference. In Brunner et al. (2020) a similar analysis is conducted for the Ethereum public blockchain, which is used as the basis for the prototypical implementation in this paper.

Generally, current blockchain implementations such as Ethereum or Bitcoin do not provide any (hard) real-time guarantees (Croman et al. 2016). The speed of processing rather depends on the load in the network, the fees associated with a transaction, and lastly on the choice of the miners.

Given a reasonable transaction fee, however, a transaction takes approximately 15 to 30 s to be processed by the Ethereum network (Vujičić et al. 2018). Most commonly. the electricity market is trading in resolutions of 15 min or less in a day-ahead market (Bathurst et al. 2002; Pinson et al. 2007). A latency of around 30 s is therefore sufficient for trading the certificates in the proposed use case.

Table 4 Complexity analysis of operations in the Event State Machine when executing the GECKO protocol

Event	Optimized	Naïve
$E_{Publish}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{A}\rvert)$
$E_{Certify}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{A}\rvert + \lvert\mathbb{E}\rvert)$
$E_{Produce}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{E}\rvert)$
$E_{Consume}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{E}\rvert)$

Table 5 Complexity analysis of operations in the Event State Machine when executing the GECKO protocol with confirmed DSO networks

Event	Optimized	Naïve
$E_{Confirm}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{E}\rvert + (\lvert\mathbb{C}\rvert + \lvert\mathbb{P}\rvert)^2)$
$E_{Consume}$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert\mathbb{E}\rvert)$

Conclusion

We presented GECKO, a blockchain-based system for green energy certificates. It is capable of representing verifiable green energy production and consumption through DSO-certified sustainable energy resources with kWh granularity. We showed that the proposed system is capable of (i) green energy trading beyond DSO borders through a confirmation-based system; (ii) scaling to trade within even a very large geographic region with a high percentage of renewable energy resources; and (iii) being implemented with constant-time complexity for all system events. In addition, we outlined a prototypical implementation, a practical extension and detailed analysis of which is future work. Future work will also discuss the handling of transmission loss as integral part of the protocol. An aspect that is currently negotiated on the market layer.

Acknowledgements
The financial support by the Federal State of Salzburg is gratefully acknowledged. Funding by the Austrian Research Promotion Agency (FFG) under project number 865082 (ProChain) is gratefully acknowledged.

Authors' contributions
This paper was written by Fabian Knirsch (51%), Clemens Brunner (30%), Andreas Unterweger (18%) and Dominik Engel (1%). The detailed contributions are as follows: The idea of this paper was developed by Fabian Knirsch (60%) and Clemens Brunner (40%). The Abstract, the "Introduction" section and the "Related work" sections were written by Fabian Knirsch (100%). The "Preliminaries" section and the "GECKO"section were written by Fabian Knirsch (33%), Clemens Brunner (33%) and Andreas Unterweger (34%). The "Practical concerns" section was written by Fabian Knirsch (20%), Clemens Brunner (40%) and Andreas Unterweger (40%). Figures were made by Fabian Knirsch (40%), Clemens Brunner (40%) and Andreas Unterweger (20%). The formalization was developed by Fabian Knirsch (25%), Clemens Brunner (25%) and Andreas Unterweger (50%). Implementations and evaluations were made by Clemens Brunner (100%). Editorial work was done by Dominik Engel (100%).

Funding
This work is funded by the Austrian Research Promotion Agency (FFG) under project number 865082 (ProChain).

Availability of data and materials
There is no additional data or material.

Competing interests
The authors declare that they have no competing interests.

Received: 31 October 2019 Accepted: 3 February 2020
Published online: 14 February 2020

References

Bartoletti M, Pompianu L (2017) An analysis of Bitcoin OP_RETURN metadata. In: 21st International Conference on Financial Cryptography and Data Security (FC 2017). Springer, Sliema. pp 218–230. http://arxiv.org/abs/1702.01024
Bathurst GN, Weatherill J, Strbac G (2002) Trading wind generation in short term energy markets. IEEE Trans Power Syst 17(3):782–789
Benet J (2014) IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3). Technical report, IPFS. http://arxiv.org/abs/1407.3561v1. https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf. Accessed 25 Oct 2019
Brunner C. (2017) Eduthereum: A System for Storing Educational Certificates in a Public Blockchain, Master's thesis. Universität Innsbruck, Innsbruck
Brunner C, Knirsch F, Engel D (2019) SPROOF: A platform for issuing and verifying documents in a public blockchain. In: Proceedings of the 5th International Conference on Information Systems Security and Privacy. SciTePress, Prague. pp 15–25
Brunner C, Knirsch F, Engel D (2020) SPROOF: A Decentralized Platform for Attribute-based Authentication. Communications in Computer and Information Science Series Book, Springer
Christidis K, Devetsikiotis M (2016) Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4:2292–2303
Conti M, Poovendran R, Secchiero M (2012) FakeBook: Detecting fake profiles in on-line social networks. IEEE. https://doi.org/10.1109/asonam.2012.185

Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) Introduction To Algorithms, 2nd edn.. MIT Press, Cambridge
Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba A, Miller A, Saxena P, Shi E, Gün Sirer E, Song D, Wattenhofer R (2016) On Scaling Decentralized Blockchains. In: International Conference on Financial Cryptography and Data Security. Springer, Christ Church. pp 106–125
Danezis G, Mittal P (2009) SybilInfer: Detecting Sybil Nodes using Social Networks. In: Network & Distributed System Security Symposium (NDSS). Internet Society, San Diego
Douceur JR (2002) The Sybil Attack. In: Druschel P, Kaashoek F, Rowstron A (eds). Peer-to-Peer Systems, IPTPS 2002. Lecture Notes in Computer Science. Springer, Berlin Vol. 2429. pp 251–260
European Commission (2012) 2012/148/EU: Commission Recommendation of 9 March 2012 on preparations for the roll-out of smart metering systems. http://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32012H0148. Accessed 25 Oct 2019
European Comm (2017) Directive of the European Parliament and of the Council on the promotion of the use of energy from renewable sources (recast). https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52016PC0767R(01). Accessed 25 Oct 2019
Han R, Gramoli V, Xu X (2018) Evaluating Blockchains for IoT. In: 9th IFIP Conference on New Technologies, Mobility & Security (NTMS 2018). IEEE/IFIP, Paris. pp 0–4
Hustveit M, Frogner JS, Fleten SE (2017) Tradable green certificates for renewable support: The role of expectations and uncertainty. Energy 141:1717–27
Ilic D, Da Silva PG, Karnouskos S, Griesemer M (2012) An energy market for trading electricity in smart grid neighbourhoods. In: IEEE International Conference on Digital Ecosystems and Technologies. IEEE, Atlanta. pp 1–6
Knirsch F, Unterweger A, Engel D (2018) Privacy-preserving Blockchain-based Electric Vehicle Charging with Dynamic Tariff Decisions. J Comput Sci Res Dev (CSRD) 33(1):71–79
Mashhour E, Moghaddas-Tafreshi SM (2010) Bidding Strategy of Virtual Power Plant for Participating in Energy and Spinning Reserve Markets — Part I : Problem Formulation. IEEE Trans Power Syst 26(2):949–956
Mengelkamp E, Gärttner J, Rock K, Kessler S, Orsini L, Weinhardt C (2018a) Designing microgrid energy markets: A case study: The Brooklyn Microgrid. Appl Energy 210:870–880
Mengelkamp E, Notheisen B, Beer C, Dauer D, Weinhardt C (2018b) A blockchain-based smart grid: towards sustainable local energy markets. Comput Sci Res Dev 33(1):207–214
Menezes AJ, van Oorschot PC, Vanstone SA (2001) Handbook of Applied Cryptography. 5th edn.. CRC Press, Boca Raton
Mihaylov M, Jurado S, Avellana N, Van Moffaert K, De Abril IM, Nowé A (2014a) NRGcoin: Virtual currency for trading of renewable energy in smart grids. In: 11th International Conference on the European Energy Market, EEM. IEEE, Krakow
Mihaylov M, Jurado S, Moffaert KV, Now A (2014b) NRG-X-Change - A Novel Mechanism for Trading of Renewable Energy in Smart Grids. In: Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems. SciTePress, Barcelona. pp 101–106
Monacchi A, Elmenreich W (2016) Assisted energy management in smart microgrids. J Ambient Intell Humanized Comput 7(6):901–913
Morthorst PE (2003) A green certificate market combined with a liberalised power market. Energy Policy 31(13):1393–1402
Munsing E, Mather J, Moura S (2017) Blockchains for decentralized optimization of energy resources in microgrid networks. In: 2017 IEEE Conference on Control Technology and Applications (CCTA). IEEE, Mauna Lani. pp 2164–2171
Nakamoto S (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report. https://bitcoin.org/bitcoin.pdf. Accessed 25 Oct 2019
Pinson P, Chevallier C, Kariniotakis G (2007) Trading Wind Generation From Short-Term Probabilistic Forecasts of Wind Power. IEEE Trans Power Syst 22(3):1148–1156
Ramchurn S, Vytelingum P, Rogers A, Jennings N (2011) Agent-Based Control for Decentralised Demand Side Management in the Smart Grid. In: The 10th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '11. International Foundation for Autonomous Agents and Multiagent Systems, Taipei Vol. 1. pp 5–12
Sikorski JJ, Haughton J, Kraft M (2017) Blockchain technology in the chemical industry: Machine-to-machine electricity market. Appl Energy 195:234–246
Technische Richtlinie BSI TR-03109 (2015) vom 11.11.2015, Bundesamt für Sicherheit in der Informationstechnik. Bundesamt für Sicherheit in der Informationstechnik. https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03109/index_htm.html. Accessed 25 Oct 2019
The European Parliament and the Council of the European Union (2009) Directive 2009/28/EC of the European Parliament and of the Council of 23 April 2009 on the promotion of the use of energy from renewable sources and amending and subsequently repealing Directives 2001/77/EC and 2003/30/EC. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex%3A32009L0028
Vujičić D, Jagodić D, Randić S (2018) Blockchain technology, bitcoin, and Ethereum: A brief overview. In: 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH). IEEE. pp 1–6. https://doi.org/10.1109/infoteh.2018.8345547
Wirth H (2018) Aktuelle Fakten zur Photovoltaik in Deutschland. Technical Report 88. Fraunhofer-Institut für Solare Energiesysteme ISE, Freiburg
Wood G (2017) Ethereum: A Secure Decentralised Generalised Transaction Ledger. Technical report, Ethereum. http://arxiv.org/abs/arXiv:1011.1669v3. https://ethereum.github.io/yellowpaper/paper.pdf. Accessed 25 Oct 2019
Wüst K, Gervais A (2018) Do you Need a Blockchain? In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug. IEEE. pp 45–54
Yu H, Gibbons PB, Kaminsky M, Xiao F (2008) SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In: IEEE Symposium on Security and Privacy, 2008 (SP 2008). IEEE, Oakland. pp 3–17
Zhumabekuly Aitzhan N, Svetinovic D (2016) Security and Privacy in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams. IEEE Trans Dependable Secure Comput 15(5):840–852

Publisher's Note