energies

MDPI

Article

P2PEdge: A Decentralised, Scalable P2P Architecture for Energy Trading in Real-Time

Jan Kalbantner [1,*], Konstantinos Markantonakis [1], Darren Hurley-Smith [1] and Raja Naeem Akram [2] and Benjamin Semal [1]

[1] Information Security Group, Royal Holloway, University of London, Egham TW20 0EX, UK; K.Markantonakis@rhul.ac.uk (K.M.); Darren.Hurley-Smith@rhul.ac.uk (D.H.-S.); Benjamin.Semal.2018@live.rhul.ac.uk (B.S.)

[2] Department of Computer Science, University of Aberdeen, Aberdeen AB24 3FX, UK; raja.akram@abdn.ac.uk

* Correspondence: jan.kalbantner.2018@live.rhul.ac.uk

Abstract: Current Peer-to-Peer (P2P) energy market models raise serious concerns regarding the confidentiality and integrity of energy consumption, trading and billing data. While Distributed Ledger Technology (DLT) systems (e.g., blockchain) have been proposed to enhance security, an attacker could damage other parts of the model, such as its infrastructure: an adversarial attacker could target the communication between entities by, e.g., eavesdropping or modifying data. The main goal of this paper is to propose a model for a decentralised P2P marketplace for trading energy, which addresses the problem of developing security and privacy-aware environments. Additionally, a Multi-Agent System (MAS) architecture is presented with a focus on security and sustainability. In order to propose a solution to DLT's scalability issues (i.e., through transaction confirmation delays), off-chain state channels are considered for the energy negotiation and resolution processes. Additionally, a STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) security analysis is conducted within the context of the proposed model to identify potential vulnerabilities.

Keywords: smart grid; security; STRIDE; decentralised P2P model; state channel; blockchain; edge computing

Citation: Kalbantner, J.; Markantonakis, K.; Hurley-Smith, D.; Akram, R.N.; Semal, B. P2PEdge: A Decentralised, Scalable P2P Architecture for Energy Trading in Real-Time. *Energies* **2021**, *14*, 606. https://doi.org/10.3390/en14030606

Academic Editor: Tamas Keviczky
Received: 15 November 2020
Accepted: 10 January 2021
Published: 25 January 2021

1. Introduction

Technological changes have induced an ongoing transition from traditional power grids to smart grids [1]. While the traditional power grid can be characterised as a centralised and mostly passive system, the smart grid can be described as a decentralised system with a two-way flow of communication and energy between customers and producers. The characteristics of each system also apply to the way data are handled. In traditional power grid systems, transactions are managed centrally for each interaction of an energy consumer and producer, which can result in scalability issues in bi-directional distributed markets. Scalability can be defined as the ability of a system to maintain its function and performance while the system grows larger [2]. However, scaling a system does not necessarily imply that a system performs well. For instance, Hines et al. [3] found that centralised control structures are inferior to local control structures. Furthermore, in terms of resilience, fault tolerance, adaptability, security and trust, the decentralised approach is seen as potentially superior [4–7].

Numerous researchers [8–11] have proposed decentralised approaches for the implementation of management, control and business processes using smart grids based on Distributed Ledger Technology (DLT). The strength of DLT and its decentralised Peer-to-Peer (P2P) structure makes it a natural choice [8,10–14] for the implementation of a smart grid marketplace and its energy systems as a Multi-Agent System (MAS) [15]. The best known DLT, blockchain, can be described as a chain of cryptographically secured,

time-stamped and immutable blocks that exist in multiple, geographically disparate and synchronised ledger nodes. All DLT systems use Byzantine Fault-Tolerant algorithms to create consensus among the nodes and therefore create a strong non-repudiation property [10].

The currently proposed P2P energy market models ignore the security and privacy of the infrastructure and architecture of the market. For example, Luo et al. [12] presented a two-layered P2P marketplace with a multi-agent trading negotiation as a first layer and a DLT settlement system as a second layer. The model mentions only the role of prosumers—consumers that can generate energy by using Distributed Energy Resources (DERs) such as photo-voltaic systems [15]—while disregarding other market participants. Additionally, the paper barely considers security measures outside of a DLT system. In parallel, Zhang et al. [9] proposed a P2P local energy market based on game theory that includes other market participants such as consumers, electricity suppliers and a community coordinator but does not consider security considerations. The work from Guerrero et al. [13] includes distributors in their P2P market while also considering that an energy transaction does not violate any constraints of the electrical grid. However, this paper has similar security shortcomings. While some works acknowledged the importance of security through the use of DLT [8,12], potential threats from the communication between entities and the system's architecture are not considered thoroughly enough. Malicious insiders, for example, could exploit their approved status [11] to manipulate other participants' statuses in the system. An adversary could use impersonation techniques to gather information about trades and other participants, or the communication lines could be used to eavesdrop, modify data and infiltrate malware.

The main goal of this paper is to propose a model for a P2P marketplace for trading energy between autonomous agents (i.e., market participants) using DLT. Subsequently, this model is referred to as P2PEdge. P2PEdge combines emerging technologies (DLT, 5G, edge computing) which allow for the provision of dynamic, scalable and sustainable MAS. The potential trading relationships among agents are defined by a DLT network through smart contracts. Smart contracts are self-executing contracts with the terms of the agreement between the buyer and the seller [16]. As the execution of a smart contract is controlled by a decentralised, distributed DLT network, its transactions are therefore trackable and irreversible [16]. However, DLT systems are subject to transaction limitations due to delays of transaction confirmations [17]. The delay results in high latency, which is unsustainable when a transaction needs to be completed in milliseconds instead of every hour [10]. An off-chain resolution is considered as it provides a verifiable, real-time solution: an off-chain solution enhances the scalability of the energy market, increasing the security and the privacy of the model [18,19].

Another issue is that a real-time marketplace comes at an additional performance cost. For instance, the installation of advanced metering infrastructure (e.g., smart meters) and energy management systems is necessary to support the communication between consumers and the grid. With smart meters aggregating a tremendous amount of data that are difficult to transfer, analyse and store [20], new solutions need to be introduced. One potential solution that can circumvent the data problem is the use of an edge computing architecture [21]; for instance, by adding cloud computing capabilities (i.e., storage, computational resources) at the edge of a radio access network. Edge computing could be able to process the data produced by smart meters and also increase transmission performance. The decreased latency for network communication would also aid the system to provide data analysis in near real-time [20]. However, it should be kept in mind that a performant network connection to all hosts would be necessary to prevent the formation of a bottleneck at the radio access network. The system then can be introduced to Artificial Intelligence (AI) in order to control the energy flow, manage the automated trading and forecast future loads. Specifically, our contributions in this paper include the following:

- We propose a novel architecture which combines emerging technologies such as 5G and edge computing to handle dynamic complex conditions better.

- Based on the alternating offers protocol [12,22], we present an algorithm for contract negotiation.
- We propose a DLT-enabled P2P marketplace model.
- In order to counter the scalability issues of DLT, we propose an off-chain state channel during the energy contract negotiation and payment processes [17].
- We perform a STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) threat analysis to illustrate the importance of security-aware environments. For the analysis, we consider the model from the perspective of a remote attacker who pursues capital gain by attacking the confidentiality, integrity or availability of the P2P model.

The remainder of the paper is organised as follows: Section 2 reviews relevant literature, Section 3 discusses the methodology, and the architecture of P2PEdge is described in Section 4. Section 5 shows details of the trading model, followed by a review of DLT in Section 6, a discussion of security and privacy is presented in Sections 7 and 8, and we conclude the paper with Section 9.

2. Related Work

The feasibility of P2P energy markets has been demonstrated in previous projects [23] and studies [9,12,13], which have all focused on different issues within the P2P market.

Many proposals include DLT systems, such as blockchain, in their models [8,10–13] to provide security for the market. However, only a few of these [10,11,24] have analysed the constraints of these blockchain systems. While blockchain certainly provides security through public-key cryptography, and the consensus algorithms add immutable transactions that prohibit fraud and double-spending [11,14], other aspects of security remain problematic for blockchain-based P2P models. Two of the major problems with blockchain technology are scalability and privacy leakage [16]. Through an increase in transactions, the validation time for transactions increases as well, both on an individual block level and system level. This increase occurs because block-sizes scale with transactions, as all transactions are stored in each node of the blockchain before they are validated. Additionally, blockchain technology leaks information regarding transactions, as all the information in the system is publicly accessible, such as the details of transactions and the balances [16,17]. One viable solution to increase the scalability of blockchain is off-chain resolution [19]; it also includes the protection of sensitive information through off-chain processing [25].

In [10], Li et al. proposed a consortium blockchain to address security problems and ensure the safety of transactions. Gai et al. [11] proposed a permissioned blockchain to address security issues by combining blockchain and edge computing. In [26], researchers utilise blockchain with multi-signatures for transaction security in a decentralised smart grid system. The proposed system uses anonymous encrypted message streams to protect against privacy leakage and to counter the need of third parties.

Outside of blockchain technology, limited research has been conducted on the security and privacy of P2P models. This includes the development of a security-aware environment and a privacy-preserving trading mechanism which still provides a certain flexibility and robustness. In Wang et al. [27], the authors obtained privacy for P2P energy trading between prosumers by "randomly" splitting up transactions. By distributing multiple parts of transactions to various prosumers, it is claimed that no-one in the system can obtain complete information regarding a prosumers' energy production or consumption. It should be noted that the CryptoNote protocol [28], used by the cryptocurrency Monero, uses the same principles to provide anonymity for transactions.

Others have proposed algorithms for privacy-preserving data aggregation [29], trading [11,26,30] and billing [30,31].

3. Methodology

In this section, the methodology for our P2PEdge model is discussed: this includes an explanation of the building blocks chosen for the model as well as the methodology behind the theoretical validation of the model.

The P2PEdge model is a marketplace architecture that makes use of several emerging technologies. We propose the use of load forecasting by calculating operational schedules from different aggregated data sources. A three-tier edge computing architecture was introduced to thwart problems that could arise with legacy systems that do not have enough processing power to calculate the operational schedule. In these cases, the data management system or the manager units can be used to do the calculations. Furthermore, the edge computing architecture helps the load balancing of the data shared between the entities as well as the management of the marketplace. Moreover, we introduced 5G technology in P2PEdge as a measure to increase the performance of the model even further. Through the use of 5G, we wanted to avoid the creation of any bottlenecks during transfer, processing or communication.

P2PEdge uses DLT (e.g., Blockchain) for marketplace trading; DLT has multiple advantages but also some drawbacks. For instance, one of the drawbacks is that the transaction confirmation delays restrict the scalability. Furthermore, in most DLT systems, personal data can be leaked, as all information is publicly available. As a solution, we proposed the use of state channels as an off-chain measure. We selected state channels instead of side-chains or sharding, as these techniques have their own drawbacks and need more research. For example, side-chains are only secure if the second chain also has enough nodes; otherwise, the security is at risk. Sharding, on the other hand, could result in management issues for data and nodes.

From a security perspective, we described a ransomware attacker that might have some financial gain in mind. We used this type of breach as it is most likely that this kind of attacker would try to exploit our system. During the modelling of P2PEdge and then performance of the security analysis, we had this attacker use-case in mind.

For the security analysis, we performed a STRIDE threat analysis. We chose STRIDE instead of other measures as it is a flexible, adaptive method that is widely used in industry to provide a common foundation for security analysis and its discussion [32].

STRIDE threat analysis includes the six categories of spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. In order to carry out the STRIDE security analysis, we followed the four steps outlined in [33]. We started with the modelling phase, in which we created the Data Flow Diagram (DFD) model of Figure 3. Afterwards, we analysed our model for possible risks, categorising—and where possible resolving—the issues prior to redefining the model. We note that this step mostly depends on the experience of the researching team as well as documentation from other industries and academia. After defining all possible outcomes, we combined the most salient of them and created Table 1 as a result. We then used a per-element approach to analyse every element of the DFD model according to the six STRIDE categories as well as the possible consequences of an adversary attacking the system. The results of the STRIDE per-element approach analysis are documented in Table 2. The third step in our STRIDE threat analysis was the threat elicitation phase. In this phase, we investigated the likely issues (technological or logical) which could precipitate the identified vulnerabilities and weighed up the risks. The fourth phase, risk management, is discussed throughout the following sections of this paper, following two threads: system integrity and confidentiality (privacy). System integrity is analysed and discussed as part of the model formalisation, while privacy is discussed in its own section prior to the conclusion.

4. The P2PEdge Model Architecture

P2PEdge is modelled as an MAS; a MAS comprises multiple agents that interact with each other to solve a problem while the agents maintain autonomy. Each agent has access

to specified capabilities along with resources that allow goal completion [15]. In this paper, the agents work together for the modelling of a multi-layered electricity market.

There are four layers to the model: (1) a DLT marketplace, (2) an off-chain state channel for the negotiation, (3) a communication layer for the grouping of entities, and (4) an electrical grid; i.e., the local power network. The schematic of the energy marketplace is illustrated in Figure 1.

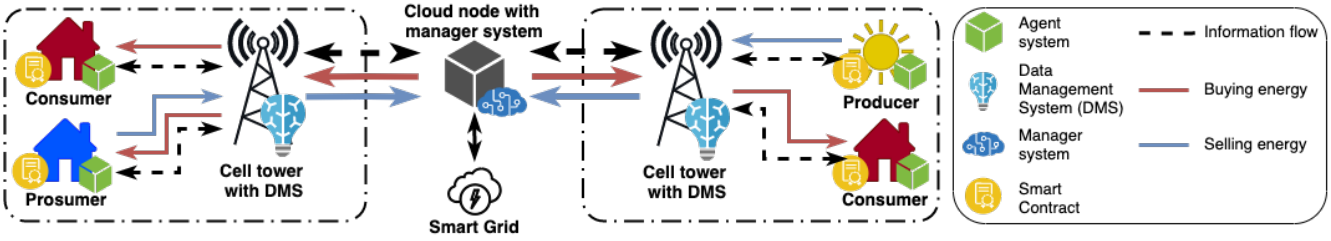

Figure 1. Proposed model architecture for P2PEdge.

4.1. Agent Preferences and Contract Selection

There are two widely known trading types in the context of a double auction. The first is named Continuous Double Auction (CDA), and other is known as Periodic Double Auction. Both variants differ in the way the auctions are handled. A Periodic Double Auction collects its bids during a specified time interval and afterwards clears the market. This variant can be used for environments in which commodities are limited and not available at all times. Furthermore, for this double auction type, the overhead is lower and, therefore, the system can be faster. However, this also means that transactions could easily break if unforeseen problems arise. Furthermore, the Periodic Double Auction could lead to drainage and increased prices. On the other hand, CDA operates by matching buyers and sellers when compatible bids are detected. Additionally, a CDA market does not hold any commodities itself, which makes it suitable for P2P markets. Furthermore, the allocation mechanisms of CDA fit decentralised markets as a centralised control is not needed. Moreover, CDA is a trading format which is widely used for, e.g., stock markets such as the NYSE. The characteristics of CDA can result in the better pricing and the better matching of buyers and sellers. However, this market format could lead to a management overheads which would increase the transaction fees in a DLT system. Moreover, the implementation of a CDA marketplace requires stricter policies regarding transactions, pricing and self-regulation [34–36].

For marketplace trading, P2PEdge uses CDA as it fits our purposes better and is expected to provide better results [34]. The proposed P2P market comprises the following:

- Trader agents $\tau \supseteq$ distributor δ, prosumer ρ, consumer γ, producer φ.
- A set of buyers S_β, where each buyer $\beta \in S_\beta$ defines his trading price P_β.
- A set of sellers S_σ, where each seller $\sigma \in S_\sigma$ defines his trading price P_σ.
- Agent systems as Home Energy Management System (HEMS), Data Management Systems (DMSs) for data collection, aggregation and group management, and a manager system as a marketplace manager (see Figure 1).
- A smart contract $SC(\cdot)$ with offers from buyers $SC(\beta, P_\beta, Q_\beta, t)$ and bids from sellers $SC(\sigma, P_\sigma, Q_\sigma, t)$.
- Energy amounts Q_β (to buy) and Q_σ (to sell) which are estimated by the agents of S_β and S_σ using the data of their smart meters.

The SC inherits the time variable t (Algorithm 1, line 4) as a measure for the DLT trading limit and is included to hinder inactivity, fraud or Denial of Service (DoS)-like attacks when trading with the off-chain state channel. Additionally, the trading timer $t^{\omega_k}_{trading}$ needs also to consider the timing limitations of the underlying DLT system (e.g., locking time of the deposit, block mining times, staking times, transmission performance, etc.). Further, we need to consider time differences that could be passed on to the marketplace

through, e.g., the temporal–spatial discretisation of power flow or an energy storage system (ESS) with longer than usual operating times. Therefore, a correction factor has to be applied to the trading timer. We note that we did not specify the trading timer mathematically, as our focus is the DLT in this paper. During an active trading session, we assume that the CDA considers each time slot individually and orders are received according to the Poisson process within a mean arrival rate λ.

Algorithm 1 Market trading buyer and seller selection procedure

procedure SELECTION PROCEDURE(S_β, S_σ) ▷ Set of Buyers & Sellers
  detect sell orders on marketplace within the group of buyers $C(\beta)$
  open state channel
  while $t < t^{\omega_k}_{trading}$ **do**
    for each $\omega_k \in \Omega$ **do**
      for each seller $\sigma \in C(\beta)$ **do**
        if $Energy^{\omega_k}_{\sigma} > 0 \vee$ criteria$X$ == *true* **then**
          $S_{SC(\beta,\sigma_k)}$ = **NEGOTIATE**(β, σ)
        end if
      end for
      if $\sigma \in \Omega \wedge \langle\beta\rangle \notin C(\beta)$ **then**
        select other group where $\langle\beta\rangle \in C(\beta)$
        $C(\beta) = C(\beta) \circ C(k)$
      else
        break
      end if
    end for
    Final contract when $S_{SC(\beta,\sigma_k)} \cup SC \wedge$ seller σ accepts $S^{final}_{SC(\beta,\sigma_k)}$
    if $\sum_{SC \in S^{final}_{SC(\beta,\sigma_k)}} SC(Q_{\beta,\sigma}) < Energy_{\sigma_k}$ **then**
      $SC^{grid}_{\beta,\sigma_k} = Energy_{\sigma_k} - \sum_{SC \in S^{final}_{SC(\beta,\sigma_k)}} SC(Q_{\beta,\sigma})$
    end if
  end while
  close state channel
end procedure

To increase privacy, only transaction-related information is exchanged among the market participants. Information related to energy consumption and production is not revealed publicly (e.g., operational status, battery status, etc.) but exchanged through encrypted channels. Before detailing our model, we will define the role and capabilities of each participant.

4.1.1. Distributors

Distributors, or Distribution System Operators (DSOs) $\delta \in \tau$, are intermediaries and own or rent part of the power grid infrastructure. They are responsible for the reliable operation of the distribution system and congestion management [15,37]. Traditionally,

the DSO is also responsible for the monitoring of flexible load operations [37]. In the model, distributors also buy and sell energy from other agents for profit. The quantity of energy bought during a given time $Q_t \in \omega^B$ must match the energy sold $Q_t \in \omega^S$. For the model, the distributor has to pay linear transaction costs to buy and sell energy, depending on the used DLT and the usage of the power grid. Accordingly, buyers and sellers will need to involve a distributor if energy is intended to be bought from a third party or when purchasing energy from the grid.

4.1.2. Prosumers

The prosumer $\rho \in \tau$ is equipped with a DER and an agent system. This agent of the prosumer can buy and sell energy in different conditions but in accordance with relevant policy. Preference-wise, the prosumer has access to DER, and, when the battery storage is replenished, the prosumer will buy energy from the marketplace or take energy from the grid. Energy usage is measured using a smart meter and may occur at any time of the day. The agent calculates the net demand by probing the smart meter data and offsetting the data against the production of energy. The battery is used to store the excess energy as well as to handle fluctuations that occur due to trading and energy transfers. An over-supply of energy will be used by the agent system to generate a selling order $\omega_k^S(\cdot) \in \Omega$, and an under-supply generates a buying order $\omega_k^B(\cdot) \in \Omega$.

4.1.3. Consumers

The consumer $\gamma \in \tau$ does not possess a DER, and therefore can only consume energy. However, the consumer can still participate in the P2PEdge market and trade contracts when equipped accordingly. In order to trade energy, the consumer needs a smart meter and a smart trading agent to control the energy consumption for the household and estimate future consumption. Furthermore, the consumer can set preferences in the marketplace for buying (e.g., buying only from certain merchants). Like the prosumer, the consumer's agent can create buying orders $\omega_k^B(\cdot) \in \Omega$.

4.1.4. Producers

Each producer $\varphi \in \tau$ owns a DER that either uses renewable energies or is based on fuel. Producers only generate energy for profit and do not want to consume energy themselves. Accordingly, they own an agent system but, in contrast to the prosumers' and consumers' agents, it is only used for trading. As with the prosumer, the producer generates selling orders $\omega_k^S(\cdot)$.

4.2. The Agent System

The agent system is a HEMS that autonomously manages the energy for a consumer, prosumer or producer. By calculating a local scheduling model, the optimal actions for the energy resources can be determined. The main objectives of the agents are energy management, autonomous trading in the marketplace and gathering information about the net demand. The monitoring of household net-demand is indispensable for advanced analytic techniques and forecast models which are used by the DSO for load shedding and load distribution (i.e., load balancing) measures across smart grids; failing to do so would risk energy fluctuations [38].

However, the limitations affecting power grid infrastructure are heterogeneous. A gate-keeping mechanism is required at critical switching stations to prevent the overloading of distribution lines. The mechanism rejects any additional throughput over a specific limit to restrict further loads from being applied.

In order to protect data privacy, a categorisation into public and private data is done which allows the agent to share personal data with other systems. For example, public data transferred to the DMS are used for advanced prediction models and energy network stability. The collected data are then analysed and further relayed. Private data are encrypted by the agent and cannot be decrypted by the DMS node. Accordingly,

an encryption model should support cryptographic tools to aggregate data confidentially. Through cryptographic tools such as additive homomorphic encryption [39], a non-trusted entity can be hindered from accessing private data [40]. The decryption key is only shared between agent and the manager systems.

In the case of a prosumer, the agent calculates the net demand by probing the internal smart meter and calculating the current demand against the energy production of any DER. For the standard consumer, the agent excludes any production from the equation. The producer only supplies energy to the grid and therefore has no demand.

The configuration of constraints is essential, as the prices in the marketplace fluctuate depending on the time of the day, weather or other unpredictable events. Thus, any owner of an edge device can set preferences and define constraints for acceptable prices as well as trading hours. By default, the agent is configured to submit offers and bids to achieve the best price possible and negotiate with every acceptable trader.

4.3. The Data Management System

The DMS manages groups of agent systems and comprises an edge computing node as well as a cellular network node. An edge node provides additional computational capabilities to achieve near real-time analysis [21]. Through the cellular node, the agent connects to the DMS. Preferably, a 5G connection is used for lower latency and better performance, but 4G is also supported for backward compatibility. Within the range of a DMS, all agent systems form a group. Member agents of the group can be added or removed at any time but can only belong to one DMS group at the same time. The combination of 5G with edge computing can provide P2PEdge with better scalability, extended computational powers and lower latency, as well as decreased response times.

The main goal of the DMS is the collection of public data from agents to forecast future loads. The analysis of current demand and supply is watched by the agents and sent through status updates to the group's DMS. A status update is used to predict future net demand. Accordingly, in regular updates, data are bundled, obfuscated for privacy-preservation and added to databases for historical data collection. The DMS will aggregate private data and regularly send them to the manager. Additionally, future demand and supply data based on public historical datasets are shared with the manager. This crucial information can help the manager system to decide specific measures; e.g., when a supply shortfall is detected, and the DSO has to get involved. Additionally, the DMS's responsibility is to relay information to the microgrid controller so that electrical contracts can be fulfilled.

4.4. The Manager System

The manager's primary purposes are data aggregation for DSOs and energy management on a regional level. Therefore, the manager is involved in the negotiation and resolution of energy contracts across multiple DMSs. The manager consolidates this information to the grid participants and distributes the energy accordingly. Considering the architecture, the manager is placed in the cloud, although it should be distributed geographically over multiple availability zones for decentralisation.

Furthermore, the manager's task is to create forecasts for the energy demand and supply of DMSs. In order to be able to do so, each DMS updates the manager with information on the group, including public data (e.g., obfuscated demand and supply information) and status updates. Status updates involve information about the group members' status, such as the number of members, consumers, prosumers and producers as well as other demographic data. However, certain circumstances require immediate status updates; for instance, when the connection to the microgrid controller is lost, or group members are disconnected without a leaving notification. Accordingly, data are shared between the agent and the manager. By encrypting private information, only authorised entities will have access. The exclusion of certain entities from access mitigates the threat of a single-point failure threat, as well as optimising the usability of resources.

Appropriate to the predicted demand, the DMS and the Transmission System Operator (TSO) will be informed. The TSO, which operates on load requests, can use this information and contact the distributor to start, e.g., load shedding or load distribution measures. When the future demand exceeds the predicted supply of nearby DMSs, the DSO distributes the loads accordingly and stores them at a nearby energy storage systems (ESS) for easier distribution to the corresponding DMS group.

4.5. Learning Model

Previously, the entities (i.e., agent systems, DMS, manager system) which can use intelligent load forecasting were discussed, where each entity uses algorithms to achieve different results. The agent system aims to automate the energy management and trade in the energy marketplace. The DMS aims to predict future loads, and the manager collects information about the loads in the case that the DSO needs to be notified and energy needs to be distributed. All of these tasks involve sophisticated AI algorithms which require learning models that are trained and back-tested. The learning model needs to fulfil certain requirements:

(1) it must be applicable to computational constraint devices;
(2) it must be adaptive—i.e., it should take various inputs (e.g., weather data) and include current consumption information;
(3) it must be able to consume large datasets (e.g., historical datasets).

Note that the underlying learning model for P2PEdge is outside the scope of this paper. Accordingly, the concrete form of the learning model depends on its implementation. For instance, a technical error in a DMS could influence the amount of gathered data which can be passed on to the DSO, which subsequently would influence the quality of the operational schedule. There are multiple models available in the literature (e.g., [41]) that can be applied. However, the outcome of the learning model is an operational schedule for the agent systems, DMSs and manager system.

5. Energy Trading Model

The core of P2PEdge is its marketplace, in which consumers, prosumers or producers can interact with each other within set boundaries and preferences. Figure 2 illustrates the workflow of the electricity trading mechanism and the workflow between the entities of the agent systems, DMS and manager system. The workflow is designed as a loop, where the agent continuously monitors the local energy resources and performs load scheduling based on the prediction model. Each time the agent updates the schedule, a message is sent to the DMS, which also updates the manager. When an agent system detects an energy shortage, it acts as a buyer in the marketplace.

The contract negotiation in the marketplace is described below. The marketplace is considered to be a materials market which solely trades energy contracts. Accordingly, when there is no successful bid, the energy will be purchased from the grid, and a process of load distribution begins. Although not depicted in Figure 2, we note that sellers can interact with the marketplace in order to generate sell offers.

Furthermore, in its current version, the model supports only atomic transactions, but it can also support the splitting of transactions into multiple packets. For the splitting of transactions, atomic transactions need to form a tree of related transactions, in such a way that the transaction at the top of the tree is the aggregate of the sub-contracts lower in the tree. Furthermore, in each branch, the terminal sub-contracts need to designate the destination or the origin for the sub-total demanded.

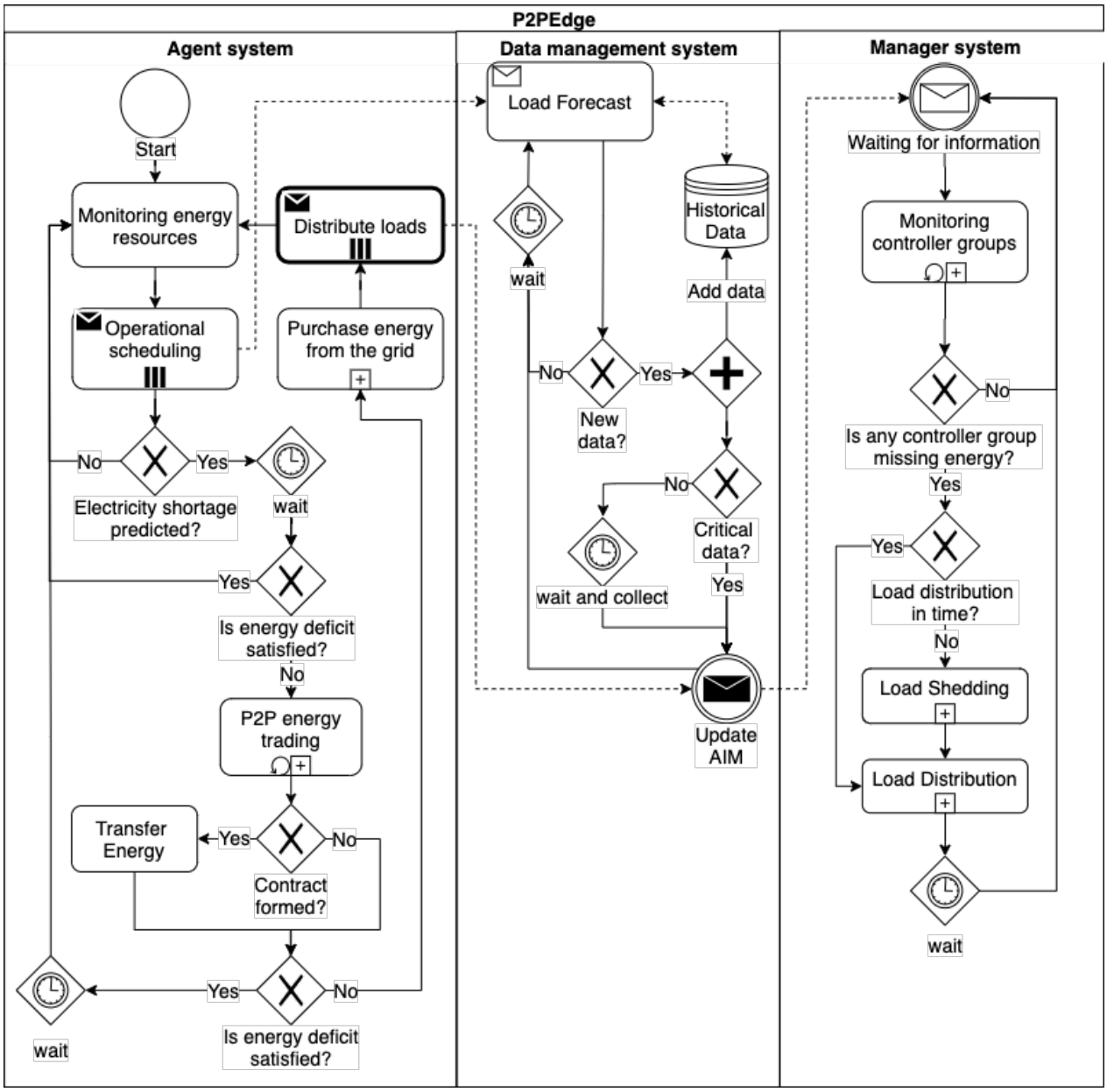

Figure 2. Workflow of the electricity trading mechanism.

5.1. Selection Procedure

Agents that detect an overproduction of energy $Energy_\sigma > 0$ will need to sell energy to the market (i.e., energy production > predicted load $\wedge$ $Energy_\sigma^{\omega_k \rightarrow ESS^{SOC}} = FULL \Rightarrow \omega_k^S$). For that purpose, an agent $\sigma \in S_\sigma$ creates an open order $\omega_k^S \in \Omega$ at the energy marketplace. The open orders are in the form of $\Omega = \Omega_1, \ldots, \Omega_m$, where $\omega_{k=1,\ldots,m}^S$ is a selling order, $\omega_{k=1,\ldots,m}^B$ is a buying order and m is the total number of orders. Every buying or selling order $\omega_k^{[B,S]}$ comprises the identity of either the buyer β or of the seller σ, the price P_β, the energy amount Q_β (in kWh) and the time t. The number of open orders is not limited in the marketplace as long as the seller σ has energy stored and can safely distribute it themselves or through a distributor.

Moreover, when the trader is a prosumer $\tau = \rho$, multiple policies could be applied when a buy or sell order should be created. For this paper, we consider the use of policies that are based on the ESS's state of charge (SOC) [42]. The SOC is a measure of the amount

of charge that is stored in a battery and, therefore, shows how much of the energy capacity of an ESS is left. The policies are displayed with the Equations (1)–(9).

For instance, if the ESS's SOC has reached the maximal capacity $Energy_{\sigma}^{\omega_k \to ESS^{SOC}} = FULL$ and the energy production is positive, a sell order can be placed ω_k^S. Therefore, energy sell orders will be fulfilled until a specified sell threshold $Energy^{\omega_k \to ESS^{SOC}_{THRES_\sigma}}$ is reached. On the other hand, if the ESS's SOC has reached the minimal capacity threshold $Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}}$ a buying order ω_k^B will be generated. The threshold for buying and selling are dependent on the maximum capacity of the ESS, the time a system would need to buy or sell the energy, transport the energy, the type of the ESS, the ESS's degradation over time and, lastly, a reserve if a problem occurs and the power grid needs to be consulted [42]. Although there are various types of ESS (e.g., mechanical, chemical, electrical and thermal), electrochemical ESSs have been shown to be widely applicable for various purposes [43].

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC}} = FULL \wedge Energy_{\sigma}^{\omega_k} > 0 \Longrightarrow \omega_k^S \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}} \quad (1)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC}} = FULL \wedge Energy_{\sigma}^{\omega_k} \leq 0 \Longrightarrow \text{No } \omega_k^{[B,S]} \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}} \quad (2)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC}} \neq FULL \wedge Energy_{\sigma}^{\omega_k} \leq 0 \Longrightarrow \text{No } \omega_k^{[B,S]} \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}} \quad (3)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC}} \neq FULL \wedge Energy_{\sigma}^{\omega_k} > 0 \Longrightarrow \omega_k^B \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\sigma}} \quad (4)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC} \mapsto t_{24}} = EMPTY \Longrightarrow \omega_k^B \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\sigma}} \quad (5)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC} \mapsto t_{24}} \neq EMPTY \wedge Energy_{\sigma}^{\omega_k \mapsto t_{24}} > 0 \Longrightarrow \text{No } \omega_k^{[B,S]} \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_{[\beta,\sigma]}}} \quad (6)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC} \mapsto t_{24}} \neq FULL \wedge Energy_{\sigma}^{\omega_k \mapsto t_{24}} > 0 \Longrightarrow \omega_k^B \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\sigma}} \quad (7)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC} \mapsto t_{24}} = FULL \wedge Energy_{\sigma}^{\omega_k \mapsto t_{24}} > 0 \Longrightarrow \omega_k^S \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}} \quad (8)$$

$$\text{If } Energy_{\sigma}^{\omega_k \to ESS^{SOC} \mapsto t_{24}} = FULL \wedge Energy_{\sigma}^{\omega_k \mapsto t_{24}} \leq 0 \Longrightarrow \omega_k^S \text{ until } Energy^{\omega_k \to ESS^{SOC}_{THRES_\beta}} \quad (9)$$

Additionally, depending on the materials selection of an electrochemical battery, it has been shown that there are operational limits that affect the degradation of the system [44]. Therefore, the SOC should be within thresholds; for instance, when the batteries' SOC approaches the lower end of the spectrum, it will become increasingly difficult to charge, and if the battery stays too long at full capacity, it will affect the efficiency and the lifetime of the battery.

Moreover, price predictions and possible price fluctuations could be made by agents to gain an economic advantage. For example, when the ESS's SOC is at maximum but the price for energy is estimated to rise, the energy could be bought and directly consumed instead of using the ESS. However, the detailed economics of market price prediction are out of scope for this paper, but they should be considered in a future extension.

The procedure for the selection of sellers illustrated in Algorithm 1 is based on a multi-agent coalition mechanism [12]. A multi-agent coalition mechanism [12,45] refers to an MAS with cooperating agents that aim to complete a common objective. The algorithm begins with the buyer detecting sellers within the same DMS group $C(\cdot)$ in the marketplace. Thus, the process is triggered, and the buyer contacts sellers to negotiate a contract before the trading timer is reached $t_{trading}^{\omega_k}$. The trading timer is set for the whole negotiation process and thwarts agents that want to hinder trading or agents that experience problems (e.g., a dropped network connection).

Furthermore, the negotiation occurs within an off-chain state channel. When a problem occurs and a dispute is raised, a dispute timer $t_{dispute}$ is set. A dispute can be raised at each step of the negotiation by each of the involved parties. Closure by dispute would require the involvement of the blockchain and cause the parties to be written on a dispute list. After n remarks are given on the dispute list, an entity will be written onto a greylist, which is then

followed by a blacklist. Additionally, the buyer calculates the available energy $Energy_{\sigma}^{\omega_k}$ of the seller based on the capacity of the power line to ensure that $Energy_{\sigma}^{\omega} \leq \kappa_{\sigma}^{max}$. After the negotiation process (see Algorithm 2), if no problem has occurred, the state channel is closed.

Moreover, in the case that there is no energy selling order available in the same DMS group, the buyer propagates the request to neighbours across different DMSs when (1) $t < t_{trading}^{\omega_k}$; (2) there is at least one agent which is not part as the group $C(\beta)$; or (3) the preferences of the local agent are configured to allow trading across multiple DMSs. After the deadline t has passed, the buyer β determines the final contract $S_{SC(\beta,\sigma_k)}^{final}$ from its temporary contracts $S_{SC(\beta,\sigma_k)}$. Furthermore, the amount of energy purchased from the grid has to be determined ($SC_{\beta,\sigma_k}^{grid}$).

Algorithm 2 Market trading negotiation procedure

procedure NEGOTIATE(β, σ) ▷ Buyer and Seller
  for order $\omega_k(\cdot)$; σ generates $\omega_k^S : SC(\sigma, P_\sigma, Q_\sigma, t)$ to β **do**
    if $P_\sigma \leq P_\beta$ **then**
      $SC(\beta, P_\beta, Q_\beta, t)$ based on $\omega_k^S(\cdot)$
      $S_{SC(\beta)} = S_{SC(\beta)} \circ SC_{\beta,\sigma}(\cdot)$
      $S_{SC(\sigma)} = S_{SC(\sigma)} \circ SC_{\beta,\sigma}(\cdot)$
      return ▷ contract negotiated
    else
      β generates $\omega_k^{B*} : SC(\beta, P_\beta^*, Q_\beta^*, t^*)$ to σ
      if $P_\sigma^* \leq P_\beta^*$ **then**
        $SC^*(\sigma, P_\sigma^*, Q_\sigma^*, t^*)$ based on $\omega_k^{B*}(\cdot)$
        $S_{SC(\beta)} = S_{SC(\beta)} \circ SC_{\beta,\sigma}^*(\cdot)$
        $S_{SC(\sigma)} = S_{SC(\sigma)} \circ SC_{\beta,\sigma}^*(\cdot)$
        return ▷ contract negotiated
      else
        return no contract ▷ no contract negotiated
      end if
    end if
  end for
end procedure

5.2. Trading Negotiation

When another agent detects an energy deficit (see Figure 2), it will request energy from the marketplace. The agents trade energy based on the set preferences of the buyer and seller. These could include achieving the best price (i.e., buying at low prices and selling at high prices), but this could also include other criteria such as preferred contracts, reliability requirements or an ethical energy supply. Subsequently, when energy is available ($Energy_{\sigma}^{\omega_k} > 0$) or certain criteria X (see Algorithm 1) are met, the buyer will start a request for a certain amount of electricity Q_β from a potential seller. The seller σ will respond and start the market trading negotiation procedure. The proposed negotiation procedure illustrated in Algorithm 2 is based on an alternating offers protocol [12,22].

First, the seller σ replies to the buyer β by generating an offer ω_k^S. The set of temporary smart contracts of the buyer $S_{SC(\beta)}$ and the seller $S_{SC(\sigma)}$ inherits smart contracts $SC(\cdot)$ based on their orders of $\omega_k^{[B,S]}$. The seller will select the price which reflects the generation cost, transmission costs and the number of existing contracts the seller has:

$$P_\sigma = P_\sigma^{gen,\omega_k} + P_{\delta,t}^{trans,\omega_k} + \varsigma_\sigma \cdot |S_{SC(\sigma)}| \tag{10}$$

where P_σ^{gen,ω_k} is the cost for generating electricity for task ω_k. Its value is determined by the operation schedule. $P_{\delta,t}^{trans,\omega_k}$ is the fixed price of the transmission cost given by the distributor δ at time t. The negotiating factor ς_σ of the seller is raised in proportion to the amount of existing contracts.

The buyer receives the offer and can either accept, reject or generate a counteroffer ω_k^{B*}. As with the seller, the buyer has to calculate its bid price:

$$P_\beta = P_{\beta,t}^{market,\omega_k} + P_{\delta,t}^{trans,\omega_k} - \varsigma_\beta \cdot |S_{SC(\beta)}| \tag{11}$$

where $P_{\beta,t}^{market,\omega_k}$ is the market retail price of a signed plan by the buyer (i.e., consumer or prosumer) at a fixed time t and $P_{\delta,t}^{trans,\omega_k}$ is the transmission cost at a fixed time t. ς_β is the negotiating factor of the buyer, which decreases in proportion to the amount of contracts available.

Based on P_β, the buyer β accepts the offer if $P_\sigma \leq P_\beta$, rejects the offer if $P_\sigma > P_\beta$ and responds with a counteroffer when $P_\beta < P_\sigma < P_{\beta,t}^{market,\omega_k}$. In the case that the buyer β makes a counteroffer ω_k^{B*}, the seller σ is at liberty to either accept or reject it. He will reject the counteroffer if the energy amount of all temporary contracts is large enough and the price of the counteroffer is smaller than the price of the temporary contracts on average ($P_\sigma^* \leq P_\beta^*$). Otherwise, the seller will accept the counteroffer, and a new contract is created $SC^*(\sigma, P_\sigma^*, Q_\sigma^*, t^*)$.

6. Trading Mechanism

All agents that want to trade on the energy marketplace compose a DLT network. Further, the network uses a strict consensus with which every agent needs to comply. The consensus in the proposed model inherits a public ledger chain and a channel. A public ledger can be based on any DLT system and is non-exclusively bound to blockchain technology. As DLT inherits the issues of scalability [16], a solution to reduce the number of transactions and increase scalability is used in P2PEdge. There are multiple open proposals to blockchain's scalability issues, such as sidechains [18], sharding [46,47] and state channels [11,19]. A sidechain is an alternative blockchain which validates data from other blockchains (i.e., the mainchain). It has block producers, which decide the order of transaction occurrences, and users that publish transactions for inclusion to the chain [18,19]. Sidechains can improve the performance of the blockchain system (i.e., the mainchain) by taking over some of the burden of transaction processing. However, sidechains also create a management overhead, as an additional chain needs to be managed (e.g., it needs its own miners for a proof-of-work-based consensus algorithm to ensure its safety against attacks).

Sharding, which is widely used by distributed database systems [48], is an approach that is now proposed for blockchain. When using sharding, the entire state of the blockchain network is split into multiple partitions that are called shards. Each shard can operate almost independently as it contains its own piece of state and transaction history. Shards require cooperative behaviour when transactions are processed that affect the data on multiple shards [46,47]. The advantage of sharding is that the throughput can be improved by parallel processing transactions. However, the downside is that sharding can become a management challenge and security risk depending on the scope to which it is applied.

If, for instance, sharding is applied to the whole DLT system, the data integrity might be compromised when one shard is hijacked by an attacker (i.e., a 1% attack) [48].

On the other hand, a state channel has n parties that agree through unanimous consent to new states which are further processed off-chain. State channels are protected against the full collusion of all other parties, and in certain situations, the states can be published to the blockchain; e.g., to resolve disputes [19]. The advantage of the state channel is the ability for parties to carry out transactions among themselves. In contrary, DLT systems need to interact with the whole network after every transaction. For the setup, each participant must deposit coins in the blockchain for the channel. After locking the coins, all parties execute the state transitions and exchange signatures to authorise each of the new states. When a participant is not co-operating or is suspected of committing fraud, the state channel entrusts the underlying DLT to resolve the dispute and self-enforce the state transition [18,19]. The chosen underlying DLT system needs to be able to cope with smart contracts as they guarantee not only the safety of coins for the online participants but also the liveness of the application so that smart contract will always progress and terminate.

6.1. Ledger Technology

The most widespread DLT is blockchain [49], which is a globally distributed ledger technology that can use different algorithms to achieve consensus among all participants [16,17]. Participants are represented by distributed nodes which can store, transfer, verify and send data. Furthermore, nodes have access to the ledger, which contains the history of network transactions. The use of a consensus algorithm ensures the validity of transactions. Only when a new transaction complies with the algorithm is it regarded as valid and added as a block to a time-stamped chain of blocks in the ledger. Blockchains can have many forms, but commonly used forms include private–permissioned and public—either permissioned or permissionless—systems. While a permissionless blockchain has no restrictions on who can access or add to the blockchain, a permissoned approach is controlled and only accessible by authorised entities [50,51]. The distinction between public and private systems refers to the access to ledger information. Private blockchains are accessible by specified users, while a public system can be accessed by anyone. It should be noted that private–permissioned systems could have limitations due to their fewer nodes when it comes to scalability and specific network attacks (e.g., a 51% attack, if an attacker gets access to the network).

In P2PEdge, the ledger technology is used to establish the preliminaries of a state channel by protecting the deposits of involved entities, resolving disputes during a channel negotiation session and updating balances after the channel is closed. Adding DLT to the model enables the use of integrity protection and non-repudiation, which are properties necessary for a real-time energy marketplace where financial loss can be a consequence.

There are multiple variations of the underlying DLT system which can be used for P2PEdge. One version of the chain could be to use one or multiple permissioned chains with authorised trader accounts [11]. This system could use a delegated proof-of-stake or proof-of-authority consensus algorithm, which utilises autonomous managers to verify blocks [52]. This approach would also significantly reduce the computational burden of conventional proof-of-work-based DLT systems [14]. Moreover, possible security risks could be drastically minimised as the network is private, and joining the network needs authorisation. However, another version of the public ledger marketplace could focus on decentralisation and would use a public available DLT system, which is most likely based on a proof-of-work consensus algorithm. This system would pose other risks but is less costly and easily deployable. Furthermore, the management overhead is minimised, which could be the main focus of specific applications. Additionally, a DLT system that enhances privacy by providing anonymity, fungibility and linkability could be used to minimise the risk of private data exposure. Examples would be Monero's CryptoNote protocol [28], ring signatures [53–55] and bulletproofs (non-interactive zero-knowledge proofs) [56].

6.2. The State Channel

The off-chain state channel is an extension of the DLT-enabled marketplace which takes on the scalability issues of DLT. This means that, in our model, on-chain and off-chain transactions need to be signed similarly; otherwise, the security of the state channel would be at risk. However, the signing process for the state channel needs to be chosen carefully, as it may affect composability when considering changes to the DLT system. One advantage of state channels is the reduction of necessary on-chain transactions, which increases their scalability. Additionally, state channels enhance privacy as transactions are shared between two parties only instead of being public. For P2PEdge, all payment and negotiation transactions should be handled off-chain. The initialisation of the creation of a state channel is started by one of the parties and enforced by the marketplace.

The creation of the channel involves the smart contract of the application $SC(\cdot)$, which is responsible for instantiating the state channel contract with the list of entities $E_1 \dots E_n \subset \tau$ and the timer for the dispute $t_{dispute}$. The status of the channel is set to *ON*, meaning that all entities can authorise new states.

With the status ON, an entity E involved in $SC(\cdot)$ can propose a new state transition:

$$\psi_{i+1} = \psi^{new}(\psi_i, \dots) \tag{12}$$

where ψ_i is the current state. The state is hashed with a nonce (as protection against replay attacks)

$$\psi_{i+1}^{nonce} = nonce(\psi_{i+1}, rand_{i+1}) \tag{13}$$

and then signed

$$Sign_E = sign(\psi_{i+1}^{nonce}, i+1) \tag{14}$$

Accordingly, the entity E gathers approval from the other entities E_n by sending ψ_{i+1}^{nonce}, ψ_{i+1}, $rand_{i+1}$ and $Sign_E$. All other entities in the channel then verify the new state before they authorise it. Therefore, they need to re-compute the new state ψ_{i+1}^{*} and its hash ψ_{i+1}^{nonce*} before verifying the signature with

$$VerifySign(E, (\psi_{i+1}^{nonce}, i+1), Sign_E) \tag{15}$$

Besides, each entity signs the hash of the state $Sign_{E_n}$ for the contract of the application $SC(\cdot)$ and the contract of the state channel SC_S as well as sending the signature to the other entities. The new state is only valid when each entity has received a signature from each other entity; otherwise, a dispute process is triggered, and the execution is continued on the DLT system.

Triggering a dispute process self-enforces the dispute time period $t_{dispute_{start}} = t_{now}$; $t_{dispute_{end}} = t_{now} + t_{dispute_{\Delta}}$ and sets the status to $DISPUTE$. All entities are required to submit their latest state hash, the version of the state and a list of signatures to provide proof that the state was authorised. The contract of the state channel $SC_S(\cdot)$ inherits the last version of the hashed state ψ_i^{nonce} only if it was signed. After the dispute period (i.e., the dispute timer is passed), any entity can resolve the dispute and therefore set the status of the channel to OFF. Furthermore, a record of the dispute ($t_{dispute_{start}}, t_{dispute_{end}}, i, E_n$) is stored on a dispute list. When the dispute is resolved, the contract SC can fetch the final state and use it for trading. Otherwise, the trading will be disrupted, and the negotiation will be aborted. If an entity is placed on the dispute list more than n times, an additional remark will be made on a local greylist. The greylist is publicly available as a smart contract within the DLT system and therefore can be checked by every external entity. When trading agents continue their suspicious behaviour and are marked on the greylist repeatedly, they will be additionally added to a blacklist. Once blacklisted, there are numerous hindrances to trading in the market, as the preferences of the agent systems are pre-configured so that one cannot trade with blacklisted agents. However, this configuration can be overridden, or a more strict configuration can be applied, and greylisted traders with a specified number of remarks on the list could also be excluded from trading with an agent. The primary purpose

of the blacklist is to block traders with malicious intent in the marketplace. However, agents that are written to greylists or blacklists can also be removed depending on set preferences. It should be noted that the usage of a greylist without the ability to remove participants may facilitate DoS attacks. For instance, an attacker could intentionally fill the greylist by employing negative behaviours, which would slow down an agent's ability to validate traders until the validation is no longer possible.

Accordingly, the optimal closure of a negotiation process would be the closure of a state channel with consent by signing the negotiated final state. The last state ψ_i^{nonce} and its version i are then submitted to the public ledger chain.

7. Security

In order to examine possible threats to the system, a STRIDE threat analysis is carried out. STRIDE [57] is a recognised and comprehensive systematic approach that aids in system security on the component level through clear impact delivery for the entire system [58]. The technique is used to identify computer security threats based on six categories which form a acronym [58]:

(1) Spoofing: The breach of user's authentication information by a third party.
(2) Tampering: The adversarial alteration of system or user data.
(3) Repudiation: A not-trusted third party user engages in activities without the ability to be traced.
(4) Information disclosure: Information is exposed to unprivileged third party individuals.
(5) Denial of Service: The attacker makes the system temporarily unavailable.
(6) Elevation of privilege: Privileged access by an unprivileged third party that then has the ability to damage or destroy the entire system.

STRIDE analyses vulnerabilities against system components which could be used by an adversarial attacker to exploit a system. A standard methodology that can be applied to every system is not given, as the system is highly flexible. In the case of the current model, our STRIDE methodology is based on the steps analysed by Scandariato et al. [33].

1. Modelling: First, a decomposition of the model is necessary for the development of a Data Flow Diagram (DFD). The DFD is used to visualise internal entities and external entities (EE), processes (P), data flow (DF) and data stores (DS).
2. Categorisation: All elements identified in the modelling phase are sorted into at least one of the six threat categories.
3. Threat elicitation: The previously identified and categorised threats are investigated to reveal any possible causes of vulnerabilities.
4. Risk management: Vulnerabilities are documented to allow further measures (e.g., risk assessment) and act appropriately according to a mitigation plan.

The success of the methodology is greatly dependent on the scope and the precision of the conducted analysis, the thoroughness of the in-depth investigation and the experience of the team members.

7.1. Use Case

For our use case, we assume a remote attacker who is in pursuit of a capital gain. The attacker could be paid to cause a disruption on either part of the system or get access to sensitive information. For instance, powerful malware could be used to attack the integrity and availability of systems (e.g., to hijack control systems) [7]. More sinister attackers may also use ransomware to extort targets to pay a ransom; in either case, the availability of the system would be attacked [59,60].

7.2. Threat Modelling

The first step is the system decomposition and the modelling of a DFD. We identified the following components to our system: an agent subsystem, a data management subsystem, a manager subsystem and a smart grid. As described previously, the DFD visualises

each of the system components and helps to identify threats. A DFD for the complete system is provided in Figure 3.

Note that, for the STRIDE analysis, smart grids are treated as black boxes which only supply high-level information. A full STRIDE analysis of smart grid security, due to its importance and length, will be the subject of another paper. At the current stage, it can be mentioned that other works have applied STRIDE to smart grid industrial control systems [61], cyber-physical systems (e.g., microgrids) [58] and blockchain [62].

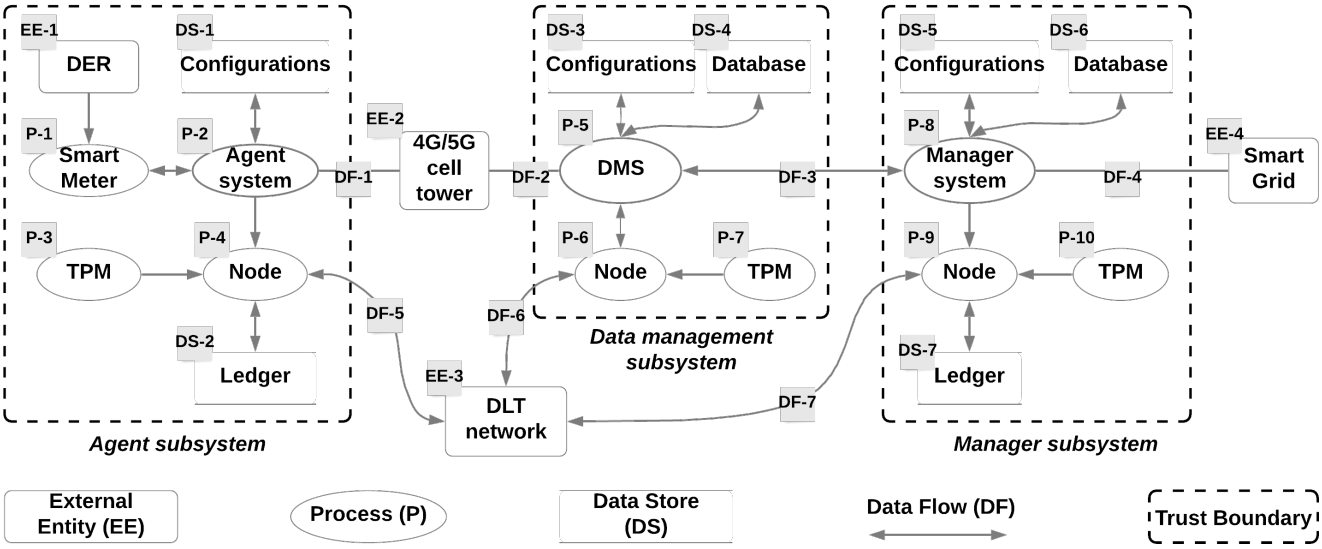

Figure 3. DFD for the model architecture. Tags are used for references.

In P2PEdge, we focus on the P2P marketplace architecture rather than on the smart grid. Additionally, we acknowledge the importance of a full smart grid security analysis and aim to research this in a future paper. In this paper, however, we focus the threat analysis on a high-level representation of other parts of the model such as the agent systems, DMS and manager system as well as the state channels.

7.3. Categorisation

For the threat analysis and categorisation (step 2), the intentions of an adversarial attacker (i.e., possible risks) have to be documented first. Table 1 documents these possible threat risks and associates each one with a risk identifier. Some of the threat risks (e.g., TR-1–TR-3) are not exclusive to the proposed model, while others are more specific (e.g., TR-8).

Table 1. Threat risks (TR) as consequences of malicious actions. DMS: Data Management System.

Risk Identifier	Description	Consequence
TR-1	Inability to communicate with agent system	-
TR-2	Inability to communicate with DMS	-
TR-3	Inability to communicate with manager system	-
TR-4	Inability to communicate with smart grid	C-2
TR-5	Inability to connect to 4G/5G cell tower	-
TR-6	Disclosure of communication secrets (e.g., keys, protocols, algorithms)	C-1
TR-7	Disclosure of system state or secrets	C-1
TR-8	Inability to meet local power demand	C-2
TR-9	Disclosure of information in the state channel	-
TR-10	Termination of state channel	-

Consequence identifiers: C-1 = financial loss, C-2 = power outage.

A STRIDE per-element approach is used that allows the separate analysis of each component's behaviour and operations. The results of the threat analysis are summarised in Table 2 and explained below.

Table 2. STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege) threat modelling using a per-element methodology. DFD: Data Flow Diagram; P: processes; EE: external entities; DF: data flow; DS: data stores.

STRIDE	DFD Elements	Threat Risks
S	P-2, P-4 - P-6, P-9, EE-2, EE-4	TR-6, TR-7
	P-2	TR-2
	P-4, P-6, P-9	TR-9
	EE-2	TR-1, TR-2
	P-5	TR-1, TR-3, TR-5
	P-8	TR-2, TR-4
	EE-4	TR-3
T	EE-1, P-1, DS-1	TR-8
	P-1, DS-3	TR-1
	P-3, P-7, P-10, DF-5 - DF-7, ... DS-2, DS-7	TR-10
	DF-1 - DF-3	TR-1, TR-2, TR-3
	DS-1 - DS-3	TR-5
	DF-4, DS-3	TR-3
	DS-1	TR-2, TR-5
	DS-5	TR-2, TR-4
R	P-4, P-6, P-9	TR-10
	P-2, P-5	TR-3, TR-7
	P-8	TR-4, TR-7
I	DS-1 - DS-3, DS-5, DS-7, P-2, ... P-5, P-8, EE-2, EE-4	TR-6
	DS-1, DS-3, DS-4, DS-6, EE-4, ... DF-1 - DF-4	TR-7
	P-4, DF-5	TR-9
D	P-2, EE-1, DS-1	TR-8
	DS-1, P-2, P-8, DS-5	TR-2, TR-4
	DF-1, DF-2	TR-1, TR-2, TR-5
	EE-2	TR-1, TR-2
	P-5, DS-3	TR-1, TR-3 - TR-5
	DF-3	TR-2, TR-3, TR-4
	DF-4	TR-3, TR-4
	EE-4	TR-4
E	P-2	TR-2, TR-5, TR-4
	P-4, P-6, P-9	TR-10
	EE-2	TR-1, TR-2
	P-5	TR-1, TR-3 - TR-5
	P-8	TR-2, TR-4
	EE-4	TR-3, TR-4

7.3.1. Spoofing

An attacker might spoof process elements (e.g., P-2, P-5, etc.) to trick other parties to disclose information about the system, such as system-critical messages (TR-7) or keys (TR-6). Furthermore, spoofing the nodes (P-4, P-6, P-9) might disclose information about the state channel (TR-9) to an attacker. An adversary could spoof multiple nodes to provoke double-spending attacks [62], which could mean a financial loss for the victim. Additionally, an attacker could spoof the authorisation or authentication credentials to gain access to systems, which would breach confidentiality. The consequently disclosed information could be used to perform other attacks, which could lead to the inability to communicate

with other elements in certain cases. For instance, spoofing the external entity EE-2 might result in the ability to connect P-2 with P-5 over DF-1 and DF-2. The results could be noticeable in the availability of the system when legitimate entities are unable to connect with, e.g., the agent (TR-1), the DMS (TR-2) or the cell tower (TR-5).

7.3.2. Tampering

In Figure 3, alterations of data or alteration attempts which could take place at multiple elements are shown. The DLT system and the state channel could be targeted (P-4, P-6, P-9, DS-2, DS-7). For instance, attackers could try to tamper with the public ledger to make a financial profit or cause damage to participants. While the consensus mechanism would hinder attackers from altering data successfully, an adversary could try to alter transactions or leak personal data. Therefore, the usage of state channels is proposed as an extension to the DLT system for better transaction protection as well as increased scalability.

An adversary could launch a tampering attack against other parts of the system. For example, P-2 makes predictions based on multiple datasets dependent on readings from the P-1. Thus, an attacker could alter information on any of the communication channels (e.g., by using a false data injection [63]).

Furthermore, EE-2 is susceptible to attacks as the external entity uses 4G/5G technology. Accordingly, the security of the technology cannot be influenced by our model. Zero-day exploits could be used for attacks and disclose a vulnerability.

7.3.3. Repudiation

Although mutual non-repudiation would be ideal, it cannot be guaranteed for every element in the DFD. Non-repudiation should be issued for critical messages, as otherwise an injection of these messages could be used by an attacker. For example, it is crucial to secure the integrity of the trading system (P-4, P-6, P-9), as the termination of the state channel (TR-10) could otherwise be risked. Furthermore, an injection of messages to P-2, P-5 or P-8 could result in the disclosure of system states or the inability to communicate with other elements.

7.3.4. Information Disclosure

Confidentiality attacks can lead to the disclosure of information which could be exploited by an attacker. For instance, the leakage of metadata is only a threat in the case of eavesdropping on DF-1–DF-4, as all other transferred data should be encrypted. Attacking the configurations (DS-1, DS-3 or DS-5) would be of more interest when an attacker aims to gather information for a more complex attack. Otherwise, P-2, P-5 or P-8 could disclose personal data or DMS group information. EE-2 could provide an attacker with more metadata about the connected parties, P-4 or DF-5 may be used to disclose information about trades on the P2P marketplace, and DS-2/DS-7 could leak key secrets of the corresponding ledgers (e.g., private keys).

7.3.5. Denial of Service

DoS attacks could interrupt the power supply, the availability of parts of the infrastructure or hinder marketplace trading. For example, if P-2 is not available, the agent system cannot trade in the marketplace and has to purchase energy from the grid. Additionally, the data for the calculation of the operational schedule will not be as detailed, even though data from different sources will be available. Furthermore, attacking P-5 will affect the whole DMS group, as status updates can no longer be received. The trading will be available for group entities, as transactions are made on the DLT system; i.e., a system with a high resistance against DoS. Moreover, P-8's availability affects the accuracy of demand and supply information. Otherwise, any estimations of energy demand and supply have to be done by the TSO, DSO and the microgrid controller. Their estimations are not as precise but will ensure the feasibility of control measures until P-8 is available again.

7.3.6. Elevation of Privilege

Elevation of privilege describes a situation in which non-authorised users can execute tasks that are beyond their privilege level. For the thread analysis, there are no privilege levels defined. Therefore, only a legitimate element or an attacker is able to execute a process. Attacking P-2 can result in an inability to verify commands from legitimate users, which may reveal configurations to an attacker. When trading on the P2P marketplace, P-4 cannot validate the authenticity of a state channel, which could lead to a termination of the channel (TR-10) when the other party commits fraud. Consequently, the termination could also lead to insufficient local electrical power (TR-8) when the demand is not met. Furthermore, for P-2, P-5, P-8 and EE-4, a privilege escalation could mean that commands cannot be verified, which could potentially lead to system failures and result in an inability to communicate with other parties.

7.4. Threat Elicitation

The third step of our STRIDE methodology is the analysis of the categorised threats. As described, the connection between the agent and DMS uses 4G/5G protocols with SIM cards. The SIM card itself can be used as a variant of a trusted platform module (TPM) which can be probed to provide a particular amount of trust [64]. However, 4G and 5G are mutually authenticated and encrypted communication protocols; therefore, no changes to the security of the communication entities are necessary unless a new zero-day vulnerability is disclosed.

Between the agents, DMSs and manager systems, the communication channels are not defined and can be chosen by the parties themselves, but they should satisfy a high security standard. Firstly, the DMS and manager exchange long-term shared secrets through a secure key exchange protocol. These long-term shared secrets are used subsequently to derive session keys which protect communication sessions. When a communication session is closed, the session key will be dumped.

Secondly, an end-to-end encrypted communication channel can be ensured through the use of long-term shared secrets in combination with freshness, a mutual authentication challenge-response protocol and mutual entity authentication. This communication channel can also be extended to the communication between the manager system and the grid. A further advantage is that this protocol would provide viable security measures against most common attacks—e.g., man-in-the-middle, impersonation and replay attacks—which is crucial for the protection of security properties. Thirdly, the communication can be secured by a TLS session.Additionally, the agent and manager entities use a long-term shared secret to exchange encrypted data, to which the same previously described process applies. All secrets are exchanged within a key management life-cycle which is a crucial part of system, as a key can be stolen or lost.

As described, the DLT system's properties allow the immutability of transactions and protection against alterations of any kind. However, the system is not privacy-preserving, as certain information has to be shared publicly. Depending on the used DLT system, a permissioned or consortium blockchain with an authorised account-model could protect the users' privacy [10,11,24]. Furthermore, current DLT systems rely on public-key cryptography, which is safe under current circumstances. However, for a sustainable system, we cannot solely rely on the future safety of public-key cryptography based on classical computer security [65], as research into achieving quantum supremacy is ongoing [66]. Novel systems need to include either lightweight post-quantum cryptography [65], which can protect against potential threats from quantum computers, or include a privacy-preserving system such as data escrow [67], where a secure and trusted third party stores the personal data.

8. Privacy

For now, the smart contract $SC(\cdot)$ and the order $\omega(\cdot)$ contain the identifier of the buyer β or of the seller σ, depending on who created the contract. By using the identifier,

the contract or the order are clearly identifiable. Therefore, personal data of all agents are exposed to a potential adversary.

For example, public information regarding a seller could be used to draw conclusions about their day-to-day life. Regular hours of increased energy distribution could suggest that sellers use less energy themselves, which again suggests that the sellers are likely not at home. However, less malicious infringements of privacy are also easily possible. For instance, advertisement companies could use the data to gather public information on households, such as the likely size of the household (inferred through the amount of consumption) or whether a household owns an electric car (inferred by uncharacteristic spikes in energy consumption during a specific time of day). As a preventive measure, the proposed model needs to be adjusted with an extension for privacy-preserving trading and billing that does not distribute any personal data; if not, the gathered information could be misused by an adversary.

For privacy preservation in P2PEdge, two different techniques could be applied: (1) identifier encryption or (2) a Universally Unique Temporary Identifier (UUTID). Identifiers can be encrypted using any kind of encryption mechanisms. For example, using a derived session secret from shared secrets can be used to encrypt and decrypt identifiers through different entities. However, this mechanism will require an established communication channel through which an additional key can be exchanged safely.

The UUTID, on the other hand, is a temporary identifier that has no fixed associations and changes frequently. The concept of the UUTID is used in telecommunication protocols (i.e., 4G, 5G) to hide identities and thwart identity theft. Currently, 4G and 5G's Global Unique Temporary Identifier (GUTI) is an 80 bit-long core network identifier that is exchanged with IMSI, a unique permanent identifier [68]. P2PEdge could utilise either of the systems as long as particular requirements are met: (1) the energy flow must remain intact after an energy contract formation, which means that certain parties need to know where to relay the energy; (2) the identity of all market participants must be hidden from other participants at all times; (3) identity theft or impersonation must not be possible; and (4) the privacy-preserving ID must have pre-image resistance and second pre-image resistance and also be resilient against collision attacks [65] (for review, see [69]).

The only parties that know the original ID—and therefore are capable of determining the identity—of a market participant should be trusted parties such as the DMS and the manager. The agent itself generates a new ID and afterwards informs its DMS. Informing the DMS is part of a handshake protocol that the agent system and the DMS have to perform. The initial handshake comprises a long-term shared secret and the ID of the agent, which is later exchanged by the privacy-preserving ID. Furthermore, the original ID will be encrypted by the shared secret of the agent and manager, which is then sent over the DMS to the manager.

Sharing the identity with the DMS and the manager is a necessity for the management of the energy flow because a finalised contract requires the seller to communicate to coordinate the transfer of energy to the buyer and direct the funds to the seller. Accordingly, all non-crucial information or personal data shared in the marketplace should be encrypted using sessions keys which are derived from the agent's and manager's shared secret. Only the participating entities should be at discretion to view specific information in the market. This also applies to the information during a state channel negotiation. However, this system will yield implications for the hardware of the agents, and thus a requirement for a certain level of computational power arises. Legacy systems, such as Supervisory Control And Data Acquisition (SCADA) systems [15], which are used for control, monitoring and management in power grids, could have too many constraints and may not be able to verify every state shared in the state channel. Therefore, either P2PEdge should not support legacy systems or other involved systems need to vouch for the legitimacy of legacy systems' states.

9. Conclusions

In this paper, we proposed a novel model for a decentralised marketplace for trading energy that protects the security and privacy of peers during data collection, trading and billing. With the usage of DLT, the marketplace inherits its cryptographic properties of strong non-repudiation and integrity protection. To overcome some of the limitations of DLT and increase privacy, we proposed the usage of state channels for trading negotiations. Further, we identified various security requirements through scenario-based elicitation techniques and discussed the applicability of our model.

Our model relies on the capabilities of multiple agents to aggregate data. Compared with existing approaches, the proposed model architecture is dynamically adjustable to different scenarios while achieving stronger cyber-physical robustness and privacy preservation. The model is only constrained by the capabilities and transmission performances of other external entities (i.e., the cloud, DSO or TSO). Furthermore, limitations apply to the edge computing node when a high volume of information needs to be computed simultaneously. The coordination of resources could prevent a computational bottleneck. The model can be used by any market participant (e.g., consumer, producer) and is non-inclusive to prosumers.

Future research may include the provision of formal proofs and the implementation of P2PEdge to provide a better comparison and performance analysis to existing models. Furthermore, a comparison of different DLT systems for P2P energy trading is planned, as well as detailed research into the security and privacy implications of future systems utilising any DLT system. Additionally, research is planned into off-chain resolution approaches to identify the unique security and service provision challenges of dynamic energy and information supply networks.

Author Contributions: Conceptualization, J.K., D.H.-S., R.N.A., B.S. and K.M.; Formal analysis, J.K., D.H.-S. and B.S.; Investigation, J.K.; Methodology, J.K.; Project administration, K.M.; Supervision, K.M. and D.H.-S.; Validation, J.K. and D.H.-S.; Visualization, J.K.; Writing—original draft, J.K.; Writing—review & editing, J.K., K.M., D.H.-S., R.N.A. and B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart Grid—The New and Improved Power Grid: A Survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
2. Sigrist, L.; May, K.; Morch, A.; Verboven, P.; Vingerhoets, P.; Rouco, L. On Scalability and Replicability of Smart Grid Projects—A Case Study. *Energies* **2016**, *9*, 195. [CrossRef]
3. Hines, P.; Bongard, J.; Burkins, M.B. *A Scalable Approach to Smart-Grid Technology or A Smarter Smart Grid*; Technical Report; UVM, College of Engineering and Mathematical Sciences: Burlington, MA, USA, 2009.
4. Siano, P.; De Marco, G.; Rolan, A.; Loia, V. A Survey and Evaluation of the Potentials of Distributed Ledger Technology for Peer-to-Peer Transactive Energy Exchanges in Local Energy Markets. *IEEE Syst. J.* **2019**, *13*, 3454–3466. [CrossRef]
5. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain Technology in the Energy Sector: A Systematic Review of Challenges and Opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
6. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I.; et al. A View of Cloud Computing. *Commun. ACM* **2010**, *53*, 50–58. [CrossRef]
7. Kshetri, N.; Voas, J. Hacking Power Grids: A Current Problem. *Computer* **2017**, *50*, 91–95. [CrossRef]
8. Münsing, E.; Mather, J.; Moura, S. Blockchains for Decentralized Optimization of Energy Resources in Microgrid Networks. In Proceedings of the IEEE CCTA, Mauna Lani, HI, USA, 27–30 August 2017; pp. 2164–2171.

9. Zhang, M.; Eliassen, F.; Taherkordi, A.; Jacobsen, H.A.; Chung, H.M.; Zhang, Y. Energy Trading with Demand Response in a Community-Based P2P Energy Market. In Proceedings of the IEEE SmartGridComm, Beijing, China, 21–23 October 2019.
10. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inf.* **2017**, *14*, 3690–3700. [CrossRef]
11. Gai, K.; Wu, Y.; Zhu, L.; Xu, L.; Zhang, Y. Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks. *IEEE Internet Things J.* **2019**, *6*, 7992–8004. [CrossRef]
12. Luo, F.; Dong, Z.Y.; Liang, G.; Murata, J.; Xu, Z. A Distributed Electricity Trading System in Active Distribution Networks Based on Multi-Agent Coalition and Blockchain. *IEEE Trans. Power Syst.* **2019**, *34*, 4097–4108. [CrossRef]
13. Guerrero, J.; Chapman, A.C.; Verbič, G. Decentralized P2P Energy Trading under Network Constraints in a Low-Voltage Network. *IEEE Trans. Smart Grid* **2019**, *10*, 5163–5173. [CrossRef]
14. Mohsin, A.H.; Zaidan, A.A.; Zaidan, B.B.; Albahri, O.S.; Albahri, A.S.; Alsalem, M.A.; Mohammed, K.I. Blockchain Authentication of Network Applications: Taxonomy, Classification, Capabilities, Open Challenges, Motivations, Recommendations and Future Directions. *Comp. Stand. Interf.* **2019**, *64*, 41–60. [CrossRef]
15. Kantamneni, A.; Brown, L.E.; Parker, G.; Weaver, W.W. Survey of Multi-Agent Systems for Microgrid Control. *Eng. Appl. Artif. Intell.* **2015**, *45*, 192–203. [CrossRef]
16. Lin, I.C.; Liao, T.C. A Survey of Blockchain Security Issues and Challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659. [CrossRef]
17. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 11–14 December 2017; pp. 557–564.
18. Back, A.; Corallo, M.; Dashjr, L.; Friedenbach, M.; Maxwell, G.; Miller, A.; Poelstra, A.; Timón, J.; Wuille, P. *Enabling Blockchain Innovations with Pegged Sidechains*; Technical Report; Blockstream Corp. Inc.: Victoria, BC, Canada, 2014.
19. McCorry, P.; Buckland, C.; Bakshi, S.; Wüst, K.; Miller, A. *You Sank My Battleship! A Case Study to Evaluate State Channels as a Scaling Solution for Cryptocurrencies*; LNCS; Financial Cryptography and Data Security; Bracciali, A., Clark, J., Pintore, F., Rønne, P.B., Sala, M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 35–49.
20. Chin, W.L.; Li, W.; Chen, H.H. Energy Big Data Security Threats in IoT-Based Smart Grid Communications. *IEEE Commun. Mag.* **2017**, *55*, 70–75. [CrossRef]
21. Taleb, T.; Samdanis, K.; Mada, B.; Flinck, H.; Dutta, S.; Sabella, D. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1657–1681. [CrossRef]
22. An, B.; Lesser, V.R.; Irwin, D.E.; Zink, M. Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing. In Proceedings of the AAMAS, Toronto, ON, Canada, 10–14 May 2010; Volume 10, pp. 981–988.
23. Zhang, C.; Wu, J.; Long, C.; Cheng, M. Review of Existing Peer-to-Peer Energy Trading Projects. *Energy Procedia* **2017**, *105*, 2563–2568. [CrossRef]
24. Kang, J.; Yu, R.; Huang, X.; Maharjan, S.; Zhang, Y.; Hossain, E. Enabling Localized Peer-to-Peer Electricity Trading among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains. *IEEE Trans. Ind. Inf.* **2017**, *13*, 3154–3164. [CrossRef]
25. Li, C.; Palanisamy, B.; Xu, R. Scalable and Privacy-Preserving Design of On/Off-Chain Smart Contracts. In Proceedings of the 2019 IEEE ICDEW, Macao, China, 8–12 April 2019; pp. 7–12.
26. Aitzhan, N.Z.; Svetinovic, D. Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. *IEEE TDSC* **2018**, *15*, 840–852. [CrossRef]
27. Wang, Z.; Yu, X.; Mu, Y.; Jia, H. A Distributed Peer-to-Peer Energy Transaction Method for Diversified Prosumers in Urban Community Microgrid System. *Appl. Energy* **2020**, *260*, 114327. [CrossRef]
28. van Saberhagen, N. CryptoNote v 2.0. Technical Report. 2013. Available online: https://cryptonote.org/whitepaper.pdf (accessed on 16 January 2021).
29. Dimitriou, T.; Karame, G. *Privacy-Friendly Tasking and Trading of Energy in Smart Grids*; ACM SAC; ACM Press: New York, NY, USA, 2013.
30. Abidin, A.; Aly, A.; Cleemput, S.; Mustafa, M.A. Secure and Privacy-Friendly Local Electricity Trading and Billing in Smart Grid. *arXiv* **2018**, arXiv:1801.08354.
31. Jawurek, M.; Johns, M.; Kerschbaum, F. Plug-in Privacy for Smart Metering Billing. *arXiv* **2011**, arXiv:1012.2248.
32. Hussain, S.; Kamal, A.; Ahmad, S.; Rasool, G.; Iqbal, S. Threat modelling methodologies: A survey. *Sci. Int.* **2014**, *26*, 1607–1609.
33. Scandariato, R.; Wuyts, K.; Joosen, W. A Descriptive Study of Microsoft's Threat Modeling Technique. *Requir. Eng.* **2015**, *20*, 163–180. [CrossRef]
34. Izakian, H.; Abraham, A.; Ladani, B.T. An Auction Method for Resource Allocation in Computational Grids. *Future Gener. Comput. Syst.* **2010**, *26*, 228–235. [CrossRef]
35. Wang, J.; Wang, Q.; Zhou, N.; Chi, Y. A Novel Electricity Transaction Mode of Microgrids Based on Blockchain and Continuous Double Auction. *Energies* **2017**, *10*, 1971. [CrossRef]
36. Chaggar, S.; Noble, J.; Cliff, D. The Effects of Periodic and Continuous Market Environments on the Performance of Trading Agents. In Proceedings of the Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, Winchester, UK, 5–8 August 2008; MIT Press: Cambridge, MA, USA, 2008; pp. 110–117.

37. Antoniadou-Plytaria, K.E.; Kouveliotis-Lysikatos, I.N.; Georgilakis, P.S.; Hatziargyriou, N.D. Distributed and Decentralized Voltage Control of Smart Distribution Networks: Models, Methods, and Future Research. *IEEE Trans. Smart Grid* **2017**, *8*, 2999–3008. [CrossRef]
38. Omran, W.A.; Kazerani, M.; Salama, M.M.A. Investigation of Methods for Reduction of Power Fluctuations Generated From Large Grid-Connected Photovoltaic Systems. *IEEE Trans. Energy Convers.* **2011**, *26*, 318–327. [CrossRef]
39. Ozdemir, S.; Xiao, Y. Secure Data Aggregation in Wireless Sensor Networks: A Comprehensive Overview. *Comput. Netw.* **2009**, *53*, 2022–2037. [CrossRef]
40. Erkin, Z.; Troncoso-pastoriza, J.R.; Lagendijk, R.L.; Perez-Gonzalez, F. Privacy-Preserving Data Aggregation in Smart Metering Systems: An Overview. *IEEE Signal Process Mag.* **2013**, *30*, 75–86. [CrossRef]
41. Almalaq, A.; Edwards, G. A Review of Deep Learning Methods Applied on Load Forecasting. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 511–516.
42. Miao, Z.; Xu, L.; Disfani, V.R.; Fan, L. An SOC-Based Battery Management System for Microgrids. *IEEE Trans. Smart Grid* **2014**, *5*, 966–973. [CrossRef]
43. Faisal, M.; Hannan, M.A.; Ker, P.J.; Hussain, A.; Mansor, M.B.; Blaabjerg, F. Review of Energy Storage System Technologies in Microgrid Applications: Issues and Challenges. *IEEE Access* **2018**, *6*, 35143–35164. [CrossRef]
44. Soloveichik, G.L. Battery Technologies for Large-Scale Stationary Energy Storage. *Annu. Rev. Chem. Biomol. Eng.* **2011**, *2*, 503–527. [CrossRef] [PubMed]
45. Horling, B.; Lesser, V. A Survey of Multi-Agent Organizational Paradigms. *Knowl. Eng. Rev.* **2004**, *19*, 281–316. [CrossRef]
46. Al-Bassam, M.; Sonnino, A.; Bano, S.; Hrycyszyn, D.; Danezis, G. Chainspace: A Sharded Smart Contracts Platform. *arXiv* **2017**, arXiv:170803778.
47. Hellings, J.; Hughes, D.P.; Primero, J.; Sadoghi, M. Cerberus: Minimalistic Multi-Shard Byzantine-Resilient Transaction Processing. *arXiv* **2020**, arXiv:200804450.
48. Kim, S.; Kwon, Y.; Cho, S. A Survey of Scalability Solutions on Blockchain. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 October 2018; pp. 1204–1207.
49. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper (accessed on 16 January 2021).
50. Peters, G.W.; Panayi, E. Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. *Bank. Beyond Banks Money* **2016**. [CrossRef]
51. Schletz, M.; Cardoso, A.; Prata Dias, G.; Salomo, S. How Can Blockchain Technology Accelerate Energy Efficiency Interventions? A Use Case Comparison. *Energies* **2020**, *13*, 5869. [CrossRef]
52. Arasev, V. POA Network Whitepaper. 2018. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 16 January 2021).
53. Noether, S.; Mackenzie, A.; Monero Core Team. *Ring Confidential Transactions*; Technical Report MRL-0005; Monero Research Lab: 2016. Available online: https://eprint.iacr.org/2015/1098 (accessed on 16 January 2021).
54. Sun, S.F.; Au, M.H.; Liu, J.K.; Yuen, T.H. RingCT 2.0: A Compact Accumulator-Based (Linkable Ring Signature) Protocol for Blockchain Cryptocurrency Monero. In *Computer Security—ESORICS 2017*; Foley, S.N., Gollmann, D., Snekkenes, E., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10493, pp. 456–474. [CrossRef]
55. Yuen, T.H.; Sun, S.F.; Liu, J.K.; Au, M.H.; Esgin, M.F.; Zhang, Q.; Gu, D. RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security. In *Financial Cryptography and Data Security*; Bonneau, J., Heninger, N., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12059, pp. 464–483. [CrossRef]
56. Bünz, B.; Bootle, J.; Boneh, D.; Poelstra, A.; Wuille, P.; Maxwell, G. Bulletproofs: Short Proofs for Confidential Transactions and More. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, May 2018; pp. 319–338.
57. Howard, M.; LeBlanc, D. *Writing Secure Code: Practical Strategies and Proven Techniques for Building Secure Applications in a Networked World*, 2nd ed.; Microsoft Press: Redmond, WA, USA, 2003.
58. Khan, R.; McLaughlin, K.; Laverty, D.; Sezer, S. STRIDE-Based Threat Modeling for Cyber-Physical Systems. In Proceedings of the 2017 IEEE PES ISGT-Europe, Torino, Italy, 26–29 September 2017; pp. 1–6.
59. Azmoodeh, A.; Dehghantanha, A.; Conti, M.; Choo, K.K.R. Detecting Crypto-Ransomware in IoT Networks Based on Energy Consumption Footprint. *J. Ambient Intell. Hum. Comput.* **2018**, *9*, 1141–1152. [CrossRef]
60. Al-rimy, B.A.S.; Maarof, M.A.; Shaid, S.Z.M. Ransomware Threat Success Factors, Taxonomy, and Countermeasures: A Survey and Research Directions. *Comput. Secur.* **2018**, *74*, 144–166. [CrossRef]
61. Jelacic, B.; Rosic, D.; Lendak, I.; Stanojevic, M.; Stoja, S. *STRIDE to a Secure Smart Grid in a Hybrid Cloud*; Computer Security; Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinoudakis, C., Kalloniatis, C., Mylopoulos, J., Antón, A., Gritzalis, S., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 77–90.
62. Hebert, C.; Di Cerbo, F. Secure Blockchain in the Enterprise: A Methodology. *Pervasive Mob. Comput.* **2019**, *59*, 101038. [CrossRef]
63. Liang, G.; Zhao, J.; Luo, F.; Weller, S.R.; Dong, Z.Y. A Review of False Data Injection Attacks against Modern Power Systems. *IEEE Trans. Smart Grid* **2017**, *8*, 1630–1638. [CrossRef]
64. Chakraborty, D.; Hanzlik, L.; Bugiel, S. *simTPM: User-Centric TPM for Mobile Devices*; USENIX Security; USENIX Association: Santa Clara, CA, USA, 2019.

65. Gao, Y.L.; Chen, X.B.; Chen, Y.L.; Sun, Y.; Niu, X.X.; Yang, Y.X. A Secure Cryptocurrency Scheme Based on Post-Quantum Blockchain. *IEEE Access* **2018**, *6*, 27205–27213. [CrossRef]
66. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum Supremacy Using a Programmable Superconducting Processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
67. Heilman, E.; Baldimtsi, F.; Goldberg, S. Blindly Signed Contracts: Anonymous on-Blockchain and off-Blockchain Bitcoin Transactions. In *Financial Cryptography and Data Security*; Clark, J., Meiklejohn, S., Ryan, P.Y., Wallach, D., Brenner, M., Rohloff, K., Eds.; LNCS; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9604, pp. 43–60.
68. Basin, D.; Dreier, J.; Hirschi, L.; Radomirovic, S.; Sasse, R.; Stettler, V. *A Formal Analysis of 5G Authentication*; ACM CCS; ACM: Toronto, ON, Canada, 2018; pp. 1383–1396.
69. Rogaway, P.; Shrimpton, T. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Fast Software Encryption*; Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3017, pp. 371–388. [CrossRef]