DECENTRALIZING SMART ENERGY MARKETS - TAMPER-PROOF DOCUMENTATION OF FLEXIBILITY MARKET PROCESSES

Zeiselmair, Andreas[1]; Guse, Miguel[1]; Yahya, Muhammad[2]; Förster, Felix[2]; Okwuibe, Godwin[2]; Birgit Haller[3]

[1] Forschungsstelle für Energiewirtschaft e.V., Am Blütenanger 71, 80995 München, www.ffe.de, azeiselmair@ffe.de

[2] OLI Systems GmbH, Silberburgstr. 112, 70176 Stuttgart, www.my-oli.com

[3] Dr. Langniß - Energie & Analyse, Silberburgstr. 112, 70176 Stuttgart, www.energieanalyse.net

The evolving granularity and structural decentralization of the energy system leads to a need for new tools for the efficient operation of electricity grids. Local Flexibility Markets (or "Smart Markets") provide platform concepts for market based congestion management. In this context there is a distinct need for a secure, reliable and tamper-resistant market design which requires transparent and independent monitoring of platform operation. Within the following paper different concepts for blockchain-based documentation of relevant processes on the proposed market platform are described. On this basis potential technical realizations are discussed. Finally, the implementation of one setup using Merkle tree operations is presented by using open source libraries.

Das Energiesystem ist zunehmend geprägt von steigender dezentraler Erzeugung und kleinteiligen Strukturen, welche neue Herausforderungen für einen effizienten Netzbetrieb schaffen. Daher werden neue Werkzeuge, sog. Flexibilitätsmärkte benötigt, die Plattform-basiert marktbasiertes Engpassmanagement bereitstellen können. In diesem Kontext ist es notwendig ein sicheres, zuverlässiges und manipulationsresistentes Marktdesign zu gewährleisten. Daher ist eine transparente und unabhängige Überwachung des Plattformbetriebs notwendig. Im folgenden Beitrag werden verschiedene Konzepte zur Blockchain-basierten Dokumentation relevanter Prozesse auf der vorgeschlagenen Marktplattform beschrieben. Auf dieser Grundlage werden mögliche technische Umsetzungsvarianten diskutiert. Abschließend wird die Implementierung einer Variante unter Verwendung von Merkle tree Operationen anhand von Open-Source-Bibliotheken vorgestellt.

1. Introduction

The energy system is already subject to fundamental change. Increasing penetration of renewable energy combined with an increased electrification of the heat and mobility sector leads to new challenges. Finally, these aspects lead to stress on the grid infrastructure. The evolving granularity and decentralization by the growing number of units and actors makes new coordination tools necessary. So called “Local Flexibility Markets” or "Smart Markets" are platform concepts currently under development in order to efficiently operate the electricity grid. [1] [2] As their main goal is to provide new approaches for market-based congestion management there is a distinct need for reliability but also tamper-resistance, so transparency and monitoring of correct platform operation is needed. Historical incidences like [3] but also current discussions in this field of research (see [4] or [5]) prove this need. The status quo of market monitoring through authorities is mainly report-based (i.e. “EU Regulation on wholesale Energy Market Integrity and Transparency”, REMIT) making it necessary for each market participant to provide all transaction and fundamental data.

Blockchain provides specific value propositions that could cover some of these needs and provide a more automated approach. On the one hand system-inherent data integrity through tamper-proof, time-specific documentation can increase trust to these newly created platforms. On the other hand, it can provide transparency through traceability of processes. Nevertheless, it also holds drawbacks regarding privacy protection and limited scalability depending on the actual setup. Therefore, different design configurations need to be assessed for specific use cases. [6] [7] [8]

2. Smart Markets

Developing a digitalized energy system providing data and controlling flexible energy units of prosumers by Smart Meters as well as measuring the physical network state by Smart Grid technologies is already an ongoing process [9] [10]. The consequential continuation to these finally provides the possibility to establish Smart Markets in order to coordinate and allocate the available flexibility to the needs of the grid [11]. Flexibility therefore can be understood as the “technical ability of a unit to change its current and/or predicted power [P, Q]” [12] [13], [14]. These flexible energy units include for example power-to-heat, distributed energy resources or energy storage systems.

Flexibility therefore is also a commodity that can be traded. In contrast to (wholesale) electricity or balancing power markets, trading flexibility for grid relief has to consider the local component to it. Congestions manifest themselves in current overload or voltage limit violation at a specific grid point. Depending on the grid topology the loads within the network have an impact on the congested element. This fact makes them not only part of the problem but eventually also part of the solution as long as they can adjust their power consumption or generation and therefore offer their flexibility. The allocation or

matching of flexibility demand at a congested grid spot to the offered flexibility by relevant flexible energy assets therefore is done by the proposed Smart Market platform [15] [16]. Fig. 1 illustrates relevant interactions between demanders and providers of flexibility to the Smart Market platform.

Involved parties consist of flexibility providers, i.e. operators of flexible energy units that offer their flexibility to the Smart Market platform and flexibility demanders, i.e. grid operators that want to contract flexibility within their grid in order to solve (predicted) congestions. Further, there is the role of the platform operator that is responsible for the correct market processes and flexibility allocation. Finally, there is also the regulatory authority that needs to control and observe correct market operation. [17] [18]

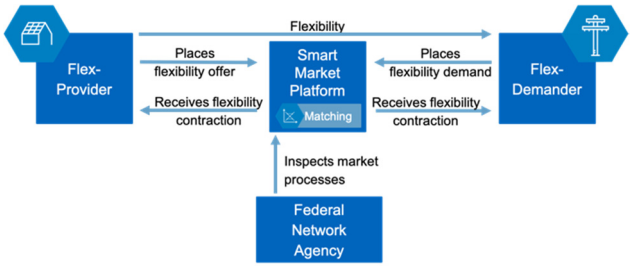

Fig. 1: Interactions of relevant Smart Market platform users

As such a Smart Market platform – as soon as in productive state – involves ten thousands up to millions of active actors (e.g. flexibility providers [19]) uploading daily datasets of relevant size and containing sensitive information, the aspects of privacy and scalability play a major role in platform design.

3. Decentralization Potentials

The decentralized character and ongoing trend in the development of the energy system involving increasing numbers of participants but also the local component of flexibility demand and provision on Smart Markets raises the question of also decentralizing the corresponding platform architecture. Taking a closer look reveals that there are three different dimensions of platform decentralization. **Operational decentralization** refers to the organizational operation of the platform. This includes both the provision of the necessary hardware and the allocation of responsibilities within the network. On the other hand **structural decentralization** addresses the platform structure itself, which is designed, for example, according to regional effectiveness, limited or defined reach or target groups. **Technical decentralization** is aimed at the actual implementation and realization of the platform or of individual functions of the platform. Different options exist starting from a jointly operated platform to complete decentralization without the need of an intermediary, e.g. using distributed ledger technology (DLT). As there is not per se an inherent value in technical decentralization it is necessary to take a closer look at potential added values provided to relevant functions, processes and finally stakeholders' needs.

4. Platform Environment incl. User Stories of Involved Parties

As part of decentralized applications, DLT in general or blockchain technology in specific aim to replace or support traditional, centralized databases, promising transparency, tamper-resistance and a high degree of availability [20], [21]. Regarding a Smart Market, this could finally lead to increasing credibility to the platform and therefore be a competitive advantage compared to alternative platform designs. Therefore, the following user stories of potential parties involved were identified regarding their need for transparency and trust:

1. **Flexibility providers** want to ensure that their flexibility offers are considered correctly on the Smart Market platform. Their demand bids should be documented immutably and time discrete to avoid conflicts. The flexibility provider should only be able to see his own offers and if applicable, corresponding contraction.
2. The **grid operator** places flexibility demands and as such wants to ensure that its demand bids are considered correctly on the market platform. The demand bids should be documented immutably and time discrete to avoid conflicts. The grid operator should only see his own demand bids, as well as (anonymized) allocated flexibility offers.
3. The **platform operator** receives flexibility demand and offers and conducts the matching algorithm. It wants to provide transparency to users by proving the correctness of registered demand and offers as well as to fulfill its reporting duties to certain authorities.
4. The **Federal Network Agency** (regulatory authority supervising electricity market) needs to ensure the correct function of the market [22]. It wants to check that all flexibility offers are considered without discrimination. Thus, it needs to be able to inspect all in- and output data (in pseudonymized form), results and version of the matching algorithm provided by the platform operator to spot-check on request.

On top of these stakeholder perspectives there are external requirements evolving from legal and regulatory frameworks. Besides safe, efficient and trusted operation one very relevant aspect is compliance of GDPR-related data privacy.

5. Concepts for Documentation of relevant Processes

In order to cover the identified needs for transparency and data security there is another challenge regarding the initial proof of correct data input. Data can be stored very securely on a blockchain but there is no impact to the correct provision of data. Especially (but not only) in the energy sector this fact shows a fundamental problem in realizing feasible end-to-end use cases. Input sources can be manifold including:

- Measurement gear that need to provide trustable sensor data to the blockchain.
- User interaction, i.e. data input coming from a user interface, e.g. providing an offer bid to a Smart Market
- External data sources, like information from third parties, e.g. weather prognosis data to a Smart Market platform.
- Computational results, i.e. solving complex problems that need to be computed off-chain, i.e. the allocation optimization of a Smart Market considering a high number of bids including constraints.

Nevertheless, there are already different approaches available to address the challenge of trusted data provision.

The most obvious approach is to regulate technical connections and the data providers themselves by a central authority. In the energy sector, available standardized and secure Smart Metering infrastructure including trusted metering point operators regulated by the Federal Network Agency and the Federal Cyber Security Authority provides a certain advantage and trust compared to other sectors [23].

A second one is to provide the possibility of checking the validity by each single user. This can be done by redundant offline storage of user-specific data and ex-post verification. This approach will be further evaluated in the following chapter. [24]

A third option is to enable different, redundant pathways to the blockchain and using consensus oracle operations as well as verifiable multi-party computation to validate the correctness of data provision [25].

Zero-Knowledge-Proofs are possibly the most elegant way of providing trusted data and especially correctly computed results without revealing all input data [26] [27]. Nevertheless, currently there are still limitations regarding scalability.

Finally, the correct application of these approaches needs to be decided on a use-case-specific point of view. Applied to the Smart Market platform an appropriate validation process could be considered in the following platform steps:

1. Provision of basic operational platform data (e.g. grid topology, boundary conditions, market area)
2. Provision of flexibility demand (by the grid operator)
3. Provision of flexibility offers (by operators of flexible energy units)
4. Optimization and provision of allocation results (through the platform-operator(s))
5. Proof of flexibility provision (through measurement data from Smart Meters)
6. Settlement information (provision of billing and payment information)
7. Revision-safe documentation

Within this paper the focus was put on the validation of flexibility offer bids (step 3) which was also realized in a proof-of-concept (see chapter 7). Besides this, the proposed setup is also applicable for steps 1 and 2.

6. Evaluation of Data Storage and Validation Options

Blockchain platforms like Ethereum provide the possibility of storing any type of data through the use of smart contracts [21]. As illustrated in Fig. 2, data can be stored openly as “plain text” within a smart contract transaction.

Storing all application data on a blockchain comes with limitations, mainly regarding scalability and data privacy. In general, scalability of blockchains is limited in terms of storage capacity and throughput. Furthermore, the cost of storage is high [28]. Current developments such as alternative consensus mechanisms, sharding or state channels aim to solve the scalability issue, but still have overhead compared to traditional databases [29], [30], [31].

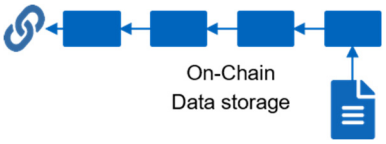

Fig. 2: On-Chain data storage

Storing private data is especially a problem on public blockchains, where data are openly accessible to anyone. Approaches to preserve confidentiality on blockchains include the use of private networks or encryption of stored data. Because encryption algorithms are susceptible to future vulnerabilities, it is questionable whether public storage of encrypted private data is compliant with regulations such as the EU’s GDPR. In addition, GDPR compliant data privacy also requires the possibility of erasing data upon request, which conflicts with the immutability of data stored on a blockchain. [32]

Considering these limitations, an alternative is to store only data hashes on-chain and storing data themselves off-chain. [33] This approach is illustrated in Fig. 3. The integrity of off-chain data can then be proven using the on-chain hash. Due to the constant length of a hash, this approach requires less storage capacity on-chain, improving scalability. The pre-image resistance of a hash function prevents private data to be inferred from its hash and thus

provides the required confidentiality [33]. As the data are stored off-chain it is also possible to erase them upon request, improving data sovereignty. Nevertheless, this approach sacrifices the blockchains improved availability and limits transparency, as data themselves are still provided off-chain.

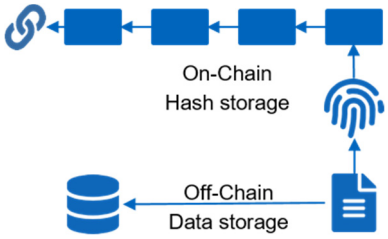

Fig. 3: On-Chain hash storage

In order to evaluate different options, relevant criteria range from privacy, scalability, accessibility, availability, data sovereignty to transaction costs depending on data volumes, as contrasted in table 1.

Table 1: Comparison of different data storage options

On-Chain storage	Plain data	Encrypted data	Hash
Scalability	×	×	✓
Privacy	×	○	✓
Data sovereignty	×	○	✓
Low transaction costs	×	×	✓
Decentralized availability	✓	✓	×
Full Transparency	✓	○	×

In order to choose a suitable approach, the requirements for the documentation of the Smart Market processes were analyzed, yielding the following results:

- Functional requirements: The data storage option must offer enough storage capacity to document the entire process and enough throughput to document it in time.
- Non-Functional requirements: Because the Smart Market also processes private data, data should be modifiable, erasable and stored confidentially, in order to comply with the GDPR. In addition, inspections by the Federal Network Agency require process data to be traceable and secured against manipulation.

Because of the scalability and privacy requirements, storing data on-chain is not an option for documentation of Smart Market processes. For this reason, the hash storage approach has been further investigated.

As an additional measure to improve scalability, process documentation data can be gathered to create a Merkle tree, resulting in a single root hash and thus less storage capacity required on the blockchain. Due to the Merkle tree's properties, this root hash alone is enough to verify the integrity of individual data entries. As mentioned in section 5, a key issue of using a blockchain for tamper-proof process documentation is to assure the correct provision of data to the blockchain. In the bidding process input data are user-provided and as such the correctness of data is determined by the user. The most efficient use of a Merkle tree structure, is to gather all input data, which however is only possible for the platform operator and not a single user. As a consequence, three different options for creating a Merkle tree and storing its root hash on a blockchain have been identified.

In the first option, illustrated in figure 4a), all Smart Market user input data for a given time frame are collected by the platform operator and then gathered to create a Merkle tree. The platform operator then stores the root hash of this Merkle tree on the Blockchain, leaving users the ability to validate the integrity of their input ex-post. With this option however, market regulators can only verify whether data supplied by the platform operator have not been manipulated since the Merkle trees creation. It is not possible to check if supplied input data are correct from a user's perspective.

In the second option, illustrated in figure 4b), platform users store their input data hash on the blockchain themselves, ensuring the correctness of the hash. Input data are supplied to the platform separately. Previous input data hashes, that are already stored on the blockchain, can be combined by the user with its own hash to create a Merkle Tree. The resulting root hash of this Merkle Tree can be stored on the blockchain by the user. This way, one root hash needs to be stored on the blockchain for each user input. Therefore, this option is less scalable as the number of transactions on-chain increases with the number of platform users.

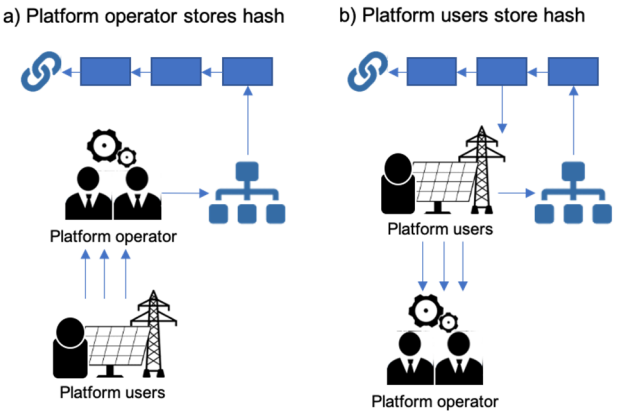

Fig. 4 a), b): Hash storage by platform operator and user

The third option, illustrated in figure 4 c), brings together both benefits of the previous options. All user input data for a given timeframe are gathered by the platform operator to create a Merkle Tree. The platform operator submits the root hash to a smart contract and requests users to verify the correctness of the root hash. A majority of users then need to sign the transaction using a multi-party consensus to

ensure its correctness, before for the smart contract stores it on the blockchain.

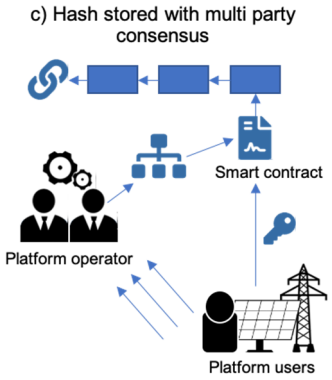

Fig. 4 c): Hash storage with multi-party consensus

While this option is scalable through its Merkle tree use and provides a check for correctness, it requires the availability of users for the consensus process. Difficulties arise from situations, where no majority consensus can be achieved or when users find their input data to have been manipulated.

Finally, the first option was chosen for the implementation described in the following section as it offers the benefits of scalability, while being more user-friendly as it requires less user interaction.

Regardless of what option would be chosen, once the root hash of a Merkle tree is stored on the blockchain it can be used for validating the integrity of data. Assuming the correctness of data used for the construction of the root hash stored on the blockchain, any data provided by the platform at a later moment can be considered untampered with, if they can be used to reconstruct an identical root hash. In the case of the aforementioned first option of storing a root hash on the blockchain, the correctness of input data used by the platform can be validated by the user ex-post. The user does this by receiving a list of hashes by the platform, which together with the user's own input data can be used to locally reconstruct the Merkle tree's root hash. If this local root hash matches the one stored on the blockchain, this proves that the user's input data have been considered correctly by the platform.

7. Implementation of Merkle tree based Proof-of-Concept

Based on the chosen concept described in the previous chapter, a proof-of-concept in the form of a prototype has been implemented for the process of provisioning flexibility offers.

In the current implementation of the ALF Smart Market, the users upload their flexibility offer as a .csv file via a dashboard on a publicly accessible domain after successful registration. The market users submit their offer one day before the actual activation of the flexibility. After the offers have been collected and the gate closure time has passed, the market operator will calculate the market result. This determines which flexibilities are being activated later on and ultimatively results in money flows.

The additional layer that is now being added to the provision process is using the blockchain along with Merkle trees. The implemented concept is illustrated in Fig. 5. In the first step, user place their offers and submit them to the platform. While the market operator is storing the data on its side, the users themselves are also creating and storing hashes corresponding to their offers locally. This technical redundancy is later used to execute the proof.

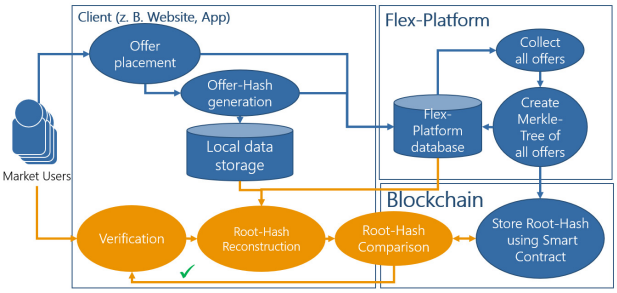

Fig. 5: Verifiable Offer Placement on Flex-Platform

After all offers have arrived at the platform, their corresponding hashes are calculated, and used to create a Merkle tree and its respective root hash. This root hash is then stored in a smart contract on the blockchain by the market operator.

Later on, in case the user wants to verify if the offer has been included correctly by the market operator, the user can request all necessary leaves for recreating the Merkle tree root hash from the platform. Using these leaves, the user can recreate a local root hash with the locally stored offer hash on the client-side. The local result can be compared with the root hash that has been stored on-chain by the market operator. This process can have the following outcomes:

- In case the root hashes match, the market operator has correctly received, included and not tampered with the offer from the user.
- In case the root hashes do not match, further investigation is required.

In theory, the approach above could be used in a diverse set of circumstances and also in other commercial sectors. In market processes, where some sort of bidding, offering or tendering involved and the market operator wants to obtain and retain a certain level of credit of trust, the operator might use this option.

As explained before, this process has been implemented as an actual technical prototype inside a standalone application. The prototype can be divided into the following components:

- **Server-side functions**, that are run at the premises or the cloud of the market operator
- **Client-side functions**, that are being executed by the browser locally on the device of the market users
- **Blockchain functions** implemented as Smart Contracts to hold the root hashes, that are being

stored by the market operator and read out by the market users

On the **client-side**, a web-application is implemented with the *vue.js* Framework. The frontend is a dashboard containing all the necessary functionality. It can be accessed via a publicly available URL. Important libraries in the web-application are:

- *Web3.js* in order to specifiy the contract address and ABI[1] of the Smart Contract. Also, web3.js features a toolset that helps connecting to a RPC-Endpoint[2] or node in order to write to or read from Smart Contracts on a designated blockchain.
- *Merkletree.js* that features a toolset to create and interact with Merkle trees and corresponding objects and attributes

Fig. 6 shows the **Create Offer Screen** which enables the user to send their offer as a .csv to the market operator. The authentication in the context of the prototype is being done with a simple username. Also, the user can select the market date, which is later used to identify which hashes from which day are supposed to be requested and compared.

The .csv file is sent to the market operator in stringified form via HTTP and a REST-API on the server.

In parallel, the hash is also created with merkletree.js and the corresponding market date is also stored in the browser storage locally for later use.

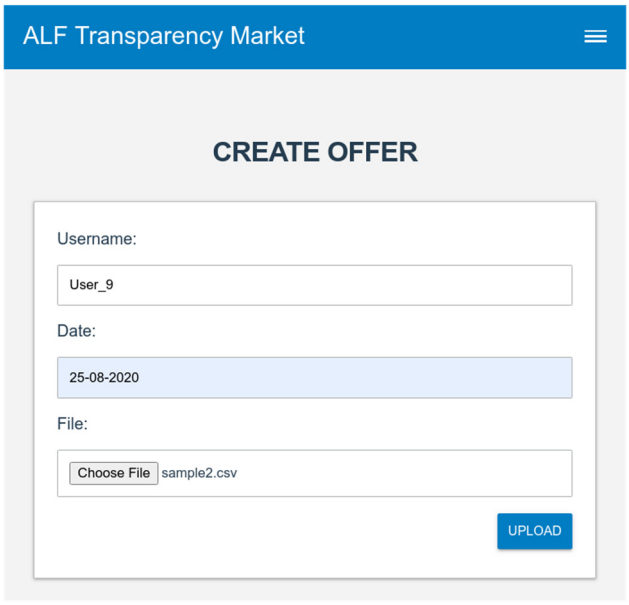

Fig. 6: Create Offer Screen for the market user

Besides the **Create Offer Screen**, there is also a need for displaying the local hashes from the browser storage. The **Show Hashes Screen** (see Fig. 7) queries the browser storage for the entries.

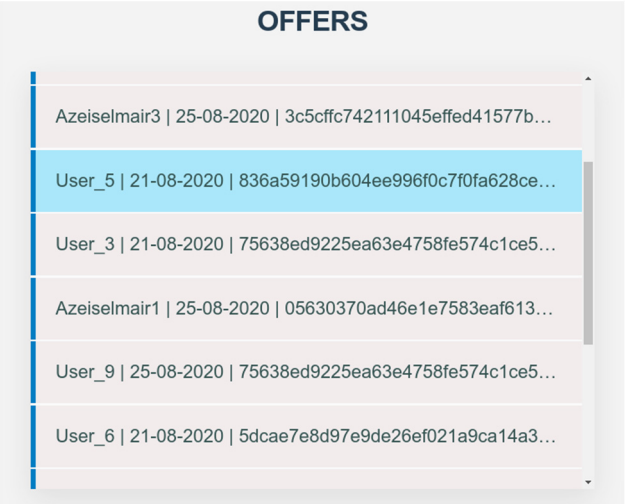

Fig. 7: Show Hashes Screen for the market user

Finally, there is also the **Verification Screen** (Fig. 8). Here, the user inputs their username and the market date. The underlying scripts will query the local hash and also request from the server, again via the REST-API, all the necessary leaves from the server in order to recalculate the hash locally. The root hash for that market date is also queried from a blockchain node and its RPC-endpoint. In case the two hashes match, a simple to understand traffic light will show the color green. In case there has been an undetermined error in the process, it will light up yellow. And in case there is a clear mismatch between the hash on the blockchain and the one created locally, the traffic light will show the color red. This concept is supposed to abstract the rather complicated processes in the background and give the user an easy-to-understand indication on the result.

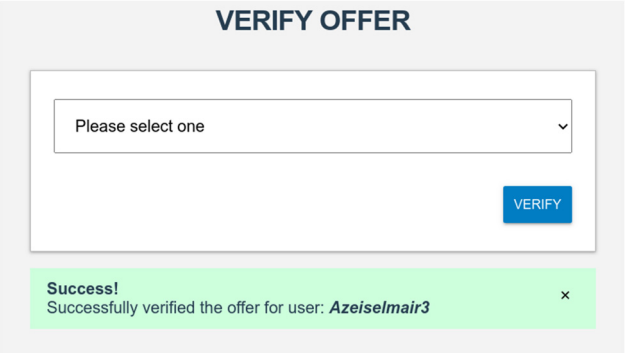

Fig. 8: Verification Screen for the market user

On the **market operator (server) side**, additional functions also have been implemented. A microservice using *node.js* as its engine is placed close to the other services that are being run by the market operator. Multiple scripts, also using the merkletree.js library, not only create the hashes, Merkle trees and root hashes from the many offers that are being received, but also provide the necessary leaves back to the users. Therefore, a database and the aforementioned *REST-API* are needed. For the prototype, we decided on *MongoDB*

[1] Application Binary Interface

[2] Remote Procedure Call

in combination with *Express.js* to operate the API, endpoint and database.

Web3.js, as realized on the client-side, is used to communicate with the blockchain. As the platform operator also has to send actual transactions to the blockchain, a key-pair with enough tokens to execute the transaction is needed.

The last and central component is the **blockchain**, which is being used in this context as a tamper-proof database that also enforces consensus. For the prototype the Volta test-blockchain was used. A public blockchain with proof-of-authority consensus algorithm (Parity Aura) operated by the Energy Web Foundation [34]. The reasons for this are:

- Only root hashes are being stored on-chain, no personal information is being put into the public domain. Therefore, the advantages of public networks can be used in full.
- PoA uses the existing extended hierarchy in the energy-sector to reduce the energy consumption in comparison to proof-of-work blockchains by orders of magnitudes. [35]

After creating a key-pair and receiving the necessary token through a faucet, the smart contract *ALFtransparency.sol* was deployed with the Truffle Framework on Volta via an open Ethereum Node operated by OLI Systems. The Contract accepts the root hashes only from the key-pair that originally deployed it. It also takes care of additional safety measures like timestamp creation on-chain.

Via the *web3.js* library, both the client-side and server-side functions can interact with the contract through an RPC, although only the market operator can successfully send transaction to it.

The whole code is open source and publicly available at *https://github.com/olisystems/alf-transparency*.

8. Critical Review and Outlook

Several blockchain-based options were analyzed as possible concepts for tamper-proof documentation of Smart Market processes with the aim of providing increased trust for platform user as well as transparency to regulatory authorities. Scalability and privacy were identified as key issues. Finally, one approach combining on- and off-chain storage using Merkle tree hashes turned out to be the most promising option, providing scalability while preserving GDPR compliant data protection. Within a proof-of-concept this approach was realized using open-source libraries including *merkletree.js*, *web3.js*, the *vue.js* frontend framework. Also, the Volta test-network from the Energy Web Foundation was chosen as the blockchain component.

Besides the achieved value propositions the following options for improvement and need for further research were identified:

- The correctness of documented data can only be verified ex-post by the users. Therefore, regulatory authorities depend on users' validation to prove the correctness of data. Data still needs to be provided by the platform operator. Already proposed possibilities of data provision using multi-party consensus could provide additional security, but further research on reducing the need for user interaction is required.
- The implementation still considers a centralized approach involving the platform operator as intermediary. Full decentralization would require a secure, scalable and privacy-preserving method for distributed computation. Current research includes the use of zero knowledge proofs or multi computation approaches solving increasingly complex calculations.
- Blockchains and their use for documentation still have to be approved by regulatory authorities as trusted resources. Therefore, further proof-of-concepts and research projects have to prove the applicability.
- Within the proposed implementation, usability was always in focus. In order to reach a productive system further automation needs to be provided.
- A detailed evaluation of synergies to other energy platforms (including smart metering infrastructure) needs to be conducted in order to reach the state of an energy business ecosystem.

Acknowledgements

The presented works are part of the project C/sells funded by the Federal Ministry of Economics and Energy (BMWi) as part of the "Schaufenster intelligente Energie - Digitale Agenda für die Energiewende" (SINTEG) funding program (funding code: 03SIN121).

References

[1] Ropenus, Stephanie: Smart-Market-Design in deutschen Verteilnetzen. Berlin: Agora Energiewende, 2017

[2] Radecke, Julia et al.: Markets for Local Flexibility in Distribution Networks - A Review of European Proposals for Market-based Congestion Management in Smart Grids. Berlin: Hertie School of Governance, 2019.

[3] Weare, Christopher: The California Electricity Crisis: Causes and Policy Options. San Francisco, California: Public Policy Institute of California, 2003.

[4] Hirth, Lion et al.: Market-Based Redispatch in Zonal Electricity Markets - Inc-Dec Gaming as a Consequence of Inconsistent Power Market Design (not Market Power). Berlin: Neon Neue Energieökonomik GmbH (Neon), 2019.

[5] Zeiselmair, Andreas et al.: Market power assessment in regional smart markets. In: 17th International Conference on the European Energy Market. Stockholm: Forschungsstelle für Energiewirtschaft e.V., 2020.

[6] Bogensperger, Alexander; Zeiselmair, Andreas, Hinterstocker, Michael: Die Blockchain-Technologie - Chance zur Transformation der Energieversorgung? - Berichtsteil Technologiebeschreibung. München: Forschungsstelle für Energiewirtschaft e.V. (FfE), 2018.

[7] Bogensperger, Alexander; Zeiselmair, Andreas; Hinterstocker, Michael; Dufter, Christa: Die Blockchain-Technologie - Chance zur Transformation der Energiewirtschaft? - Berichtsteil: Anwendungsfälle. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[8] Strüker, Jens et al.: Blockchain in der Energiewirtschaft - Potenziale für Energieversorger. Berlin: Bundesverband der Energie- und Wasserwirtschaft e.V. (BDEW), 2017.

[9] Gesetz zur Digitalisierung der Energiewende. Ausgefertigt am 2016-08-29; Berlin: BMWi, 2016.

[10] Bogensperger, Alexander; Estermann, Thomas; Samweber, Florian; Köppl, Simon; Müller, Mathias; Zeiselmair, Andreas, Wohlschlager, Daniela: Smart Meter - Umfeld, Technik, Mehrwert. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[11] Ramos, Ariana et al.: Realizing the smart grid's potential: Defining local markets for flexibility. In: Utilities Policy Vol. 40, p. 26 - 35. Michigan: Katholieke Universiteit Leuven, 2016.

[12] Müller, Mathias et al.: Dezentrale Flexibilität für lokale Netzdienstleistungen - Eine Einordnung des Flexibilitätsbegriffs als Grundlage für die Konzipierung einer Flexibilitätsplattform in C/sells. In: BWK - Das Energie-Fachmagazin 6/2018. Düsseldorf: Verein Deutscher Ingenieure (VDI), 2018.

[13] Flexibilität im Stromversorgungssystem - Bestandsaufnahme, Hemmnisse und Ansätze zur verbesserten Erschließung von Flexibilität - Diskussionspapier Stand 03. April 2017. Bonn: Bundesnetzagentur, 2017

[14] E-Bridge Consulting GmbH: Sichere und effiziente Koordinierung von Flexibilitäten im Verteilnetz. Bonn: E-Bridge Consulting GmbH, 2017.

[15] Köppl, Simon et al.: Altdorfer Flexmarkt – Decentral flexibility for distribution networks. In: Internationaler ETG-Kongress 2019. Esslingen: VDE ETG, 2019.

[16] Zeiselmair, Andreas et al.: Netzdienlicher Handel als Element des zellulären Energiesystems am Beispiel des Altdorfer Flexmarkts (ALF) - 11. Internationale Energiewirtschaftstagung (IEWT). Wien: Technische Universität Wien, 2019.

[17] Zeiselmair, Andreas et al.: Altdorfer Flexmarkt (ALF) - Konzeptbeschreibung, Zielsetzung, Funktionsweise und Prozesse des Altdorfer Flexmarkts. München: Forschungsstelle für Energiewirtschaft e.V., 2018.

[18] Zeiselmair, Andreas et al.: Altdorfer Flexmarkt (ALF) - Use Case Beschreibung. München: Forschungsstelle für Energiewirtschaft e.V., 2020.

[19] Müller, Mathias et al.: Regionales Flexibilitäts-Potenzial dezentraler Anlagen - Modellierung und Bewertung des regionalen Flexibilitäts-Potenzials von dezentralen Flexibilitäts-Typen im Verteilnetz. Berlin: Conexio GmbH, 2019.

[20] Ali, Robleh et al.: Distributed Ledger Technology: beyond block chain. London: Government Office for Science, 2016.

[21] Buterin, Vitalik: A Next Generation Smart Contract and Decentralized Application Platform - Ethereum White Paper. Switzerland: Ethereum Foundation, 2014.

[22] VERORDNUNG (EU) Nr. 1227/2011 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 25. Oktober 2011 über die Integrität und Transparenz des Energiegroß-handelsmarkts. Brüssel: Europäisches Parlament und Rat, 2011

[23] Technische Richtlinie BSI TR-03109. Bonn: Bundesamt für Sicherheit in der Informationstechnik (BSI), 2015.

[24] Eberhardt, Jacob et al.: On or Off the Blockchain? Insights on Off-Chaining Computation and Data. Berlin: TU Berlin, 2017.

[25] Aitzhan, Nurzhan et al.: Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams. Abu Dhabi: Masdar Institute of Science and Technology, 2016.

[26] Hasan, Jahid: Overview and Applications of Zero Knowledge Proof (ZKP). Nanjing: Nanjing University of Posts and Telecommunications, 2019.

[27] Ben-Sasson, Eli et al.: Scalable, transparent, and post-quantum secure computational integrity. Haifa, Israel: Zerocash, 2018.

[28] Mazlan, Ahmad et al.: Scalability Challenges in Healthcare BlockchainSystem—A Systematic Review. Kuala Lumpur: Universiti Teknologi Malaysia, 2020.

[29] Buterin, Vitalik: A Proof of Stake Design Philosophy. In: https://medium.com/@Vitalik Buterin/a-proof-of-stake-design-philosophy-506585978d51. (Abruf am 2018-01-24); (Archived by WebCite® at http://www.webcitation.org/6whn5uuin); San Francisco,

[30] Zamyatin, Alexei: On sharding blockchains. In: https://github.com/ethereum/wiki/wiki/Sharding-FAQ. (Abruf am 2018-01-01); Zug, Switzerland: Ethereum Foundation, 2017.

[31] Poon, Joseph et al.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. San Francisco: Lightning Network, 2016.

[32] Eichler, Natalie et al.: Blockchain, data protection, and the GDPR - Positionspapier. Berlin: Blockchain Bundesverband e. V., 2018.

[33] Rogaway, Phillip et al.: Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. Davis, CA: Dept. of Computer Science, University of California, 2004.

[34] Hartnett, Sam et al.: The Energy Web Chain - Accelerating the Energy Transition with an Open-Source, Decentralized Blockchain Platform. Energy Web Foundation, 2019.

[35] Sedlmeir, Johannes et al.: The Energy Consumption of Blockchain Technology: Beyond Myth. Bayreuth: Project Group Business and Information Systems Engineering of the Fraunhofer FIT, 2020.