RESEARCH

Verifiable Proofs for the Energy Supply Chain - Small Proofs Brings You a Long Way

Morten Jokumsen[1], Torben Pryds Pedersen[2], Martin Schmidt Daugaard[3], Daniel Tschudi[2], Mikkel Wienberg Madsen[4] and Thomas Wisbech[3]

Full list of author information is available at the end of the article

Abstract

We describe a solution for secure and verifiable handling of energy certificates. Such certificates are increasingly used to claim and prove responsible use of green energy, and there is a strong need for transparency and public verifiability. While the proposed solution is designed for handling electricity it applies to different types of energy as well and the concepts may also be applied to other domains. Transmission System Operators are trusted to record consumption and production of electricity. The movement from volume-based MWh yearly certificates to spot-market aligned hourly or 15 minute time-volume based intervals, creates challenges in relation to handling large amounts of data and subsequent transactions. Small discrete intervals gives the certification increased accuracy of energy consumption, as a means to prevent greenwashing, with the cost of higher amounts of transactional data and complexity. To ensure trust in the certification, these certificates must in addition be unique and publicly verifiable. This paper describes how blockchain technology can be used to create the required transparency and public verifiability. We show how large amounts of data can be efficiently handled on blockchains and how confidential data such as the amount of used energy in the certificates can be protected, ensuring privacy and correctness of the certificates.

Keywords: Management of distributed generation and demand; Granular certificates; Blockchain; Zero-knowledge proofs; Smart energy systems

Introduction

The increased focus on environmental responsibility has created a need and desire for many consumers of electricity to use green energy and have a way to prove the origin of consumed electricity. Such tracking should use the same resolution as the electricity market, currently hourly, instead of the yearly resolution used by the current Guarantee of Origin system.

This paper will devise an IT-infrastructure for auditable, verifiable and unique proofs of energy production and consumption at scale as well as consumer claims of using a specific source of (green) energy. We seek to ensure greater transparency throughout the energy supply chain by getting closer to the underlying physical properties of the electric grid and give better specificity to the choice of energy for consumers and producers. By having such infrastructure in place, there will be additional downstream benefits such as

1 Give consumers and producers verifiable proofs of their energy claims that can be audited by third parties.

2 Establish a fair marketplace for service providers that handles the certification process.
3 Higher transparency in the supply-chain and the documentation of losses upon conversion.

The goal therefore is to create a system for handling energy as a digital asset. To achieve a scalable solution, we need to address several challenges in regards to data storage, soundness of the certificates, ensure confidentiality and extend the trust that Transmission System Operators (TSOs) inherently have in the energy grid using appropriate technological solutions.

The energy space is truly data intensive and large, especially with production and consumption certificates at hourly or 15 minute intervals and to address this issue there are several strategies in play:

1 Decentralize the registries for granular certification by distributing the workload to a size where the services can run reliably.
2 Devise a strategy for each registry to be auditable and verifiable by means of applying Blockchain and Merkle trees validation (cf. Use of Blockchain) and ensuring confidentiality using zero-knowledge proofs and commitments (cf. Confidential Information).
3 Ensure that registries have a common public key infrastructure to interact with each other and validate their soundness so they can interact with stateful certificates — so the system can scale into adoption.

This paper will focus on 2. To create the transparancy needed to verify fhe state of certificates (including claims of energy from a particular producer) each registry maintains an event log of all changes to each certificate. The correctness of a certificate or claim of energy can then be verified against the event log. The solution uses a blockchain to ensure the immutability, transparency and public verifiability of the event log. For general privacy reasons or due to competition among enterprises, the amount of energy used or produced must be handled as confidential to the consumer and producer, respectively. The solution therefore hides such information in the public event log using commitments, and zero-knowledge proofs are used to prove consistency across log entries.

As the expertise and knowledge for such a solution is highly specialized and not immediately accessible to TSOs a collaborative, co-development, open source effort with partners from the blockchain and data infrastructure space was established by Energinet in a project called Project Origin [1]. It is an open and collaborative effort to solve the issue of a verifiable and trustworthy registry. Project Origin is intended as a blockchain solution that addresses the problem with high intensity data-streams and converts these data into verifiable structures with a small footprint and it can be applied within any type of supply-chain domain that seeks transparency and traceability for large amounts of transactions, including confidential data, that requires a verification layer.

The next section gives more details on the background of this work. The following sections provide an overview of the solution and provide more details on the usage of blockchain as well as the handling of private information. Terminology is explained in the text as needed; the most important terms are collected in Terminology.

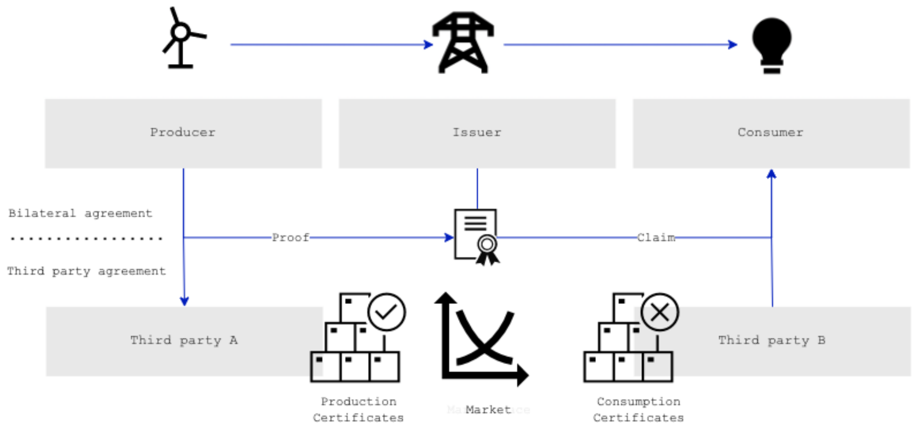

Figure 1 The market value proposition of Energy Track and Trace [2] is to provide a certification system for energy that is verifiable, auditable and unique, from producer to consumer, at spotmarket time on an hourly or 15 minute delta.

Background

Background for this Work

Energinet is the Danish TSO responsible for the national energy grids, from electricity to gas, data-acquisition etc. This unique position as a TSO in the green transition is both a big responsibility and an opportunity to service national customers and collaborate internationally with other TSO's, especially within IT-infrastructure. That is why Energinet is engaged in Energy Track and Trace (ETT) [2] which is a collaboration between TSO's Energinet (DK), Elia Group / 50 Hz (DE/BE), Elering (EE) and VertiCer (NL) to realize an infrastructure for Granular Certification that aligns with the EnergyTag [3] standards and definitions. Figure 1 gives an overview of the market value proposition of Energy Track and Trace.

Granular Certificates are accurately described by EnergyTag as:

> The central purpose of Granular Certificates (GCs) is to make electricity traceability (i.e. EACs / GCs) more closely represent the physical reality and real world availability of clean energy sources. This gives consumers the ability to demonstrate the matching of their consumption with the energy generation source of their choice on a (sub)hourly basis, or to purchase electricity at times that maximize Avoided Emissions. ([4, p. 7])

Achieving this standard is by no means trivial but by all means a necessity. Having been devised for accounting methods and purposes at a time where renewables had a marginal share of the production in the grid, the existing infrastructure unfortunately has problems such as greenwashing and lack of transparency [5, p. 7]. Times have changed and building the IT-infrastructure that can handle the data intensity, size and stateful operations in a reliable manner becomes paramount. Without appropriate documentation of the origin of the asset, in this case energy, there is no reliable or trustworthy way of accounting for consumption.

ETT seeks to extend the trust that inherently comes as a TSO, providing the energy necessary for our societies and economies to function, and leverage it to the end consumers by means of the appropriate technologies and IT-infrastructure and Project Origin is the means to this end. Providing transparency, openness, engagement and leveraging the trust from TSO's to the end consumer is the goal and obligation of granular certification using an appropriate combination of technologies.

State of the Art

In the field of energy informatics there is not much literature on similar solutions for unique and verifiable proofs at scale. There are plenty of demonstrators, prototypes and even a few proprietary systems in production using blockchains for immutability for certification throughout the supply chain of energy carriers. Such examples are [6] demonstrating the use of shielded NFTs for carbon emission tracing, the Green Tracking demonstrator from Elia Group [7] that matched production and consumption of electricity from a supplier to a consumer, Project Origin from Energinet that demonstrated the use-case up to 50 users in 2019 [8] and its implementation of Granular Certificates in Energy Origin [9]. Considering in production systems the Maersk tradelens [10] infrastructure for tracing shipping containers using blockchain technology that was discontinued in 2022 [11] showed the use and potential of commercial infrastructure for supply chain traceability in the maritime industry. The InDEED project [12] is a German project that is currently in progress and aims to demonstrate the use of blockchain technology for certification of energy carriers. There are several other potential candidates for such in production systems, however they are of proprietary nature and not easily accessible — the technology is known to be used in the industry, but the implementation details are sparse.

What they have in common is the use of decentralized blockchains or similar technologies for immutability and transparency, however they come up short due to the scale of the requirements, especially on the throughput of data required for a system to have any impact on the energy markets.

Notable issues and main obstacles making unique and verifiable digital proofs are:

- Throughput
- Scalability
- Cost
- Convenience
- Data privacy
- Trust and competition
- Energy consumption

A single metering point will be issued on hourly spot-market aligned resolution, 365 * 24 = 8760 certificates yearly, discounting trade, settlement (claim), withdraw and expire state changes on each certificate, that additionally adds to the data volumes. These requirements alone calls for a different approach than 1-1 entries on a blockchain. Sharding of blockchain processing is a method that as proposed by [13] being a method that can be used to scale the throughput of a blockchain and reduce cost and energy consumption and the method has been described by [14] to shard computing power across a network for blockhains. The method is not without its own challenges, but it is a promising approach to the problem of scaling blockchains towards real-time data throughput.

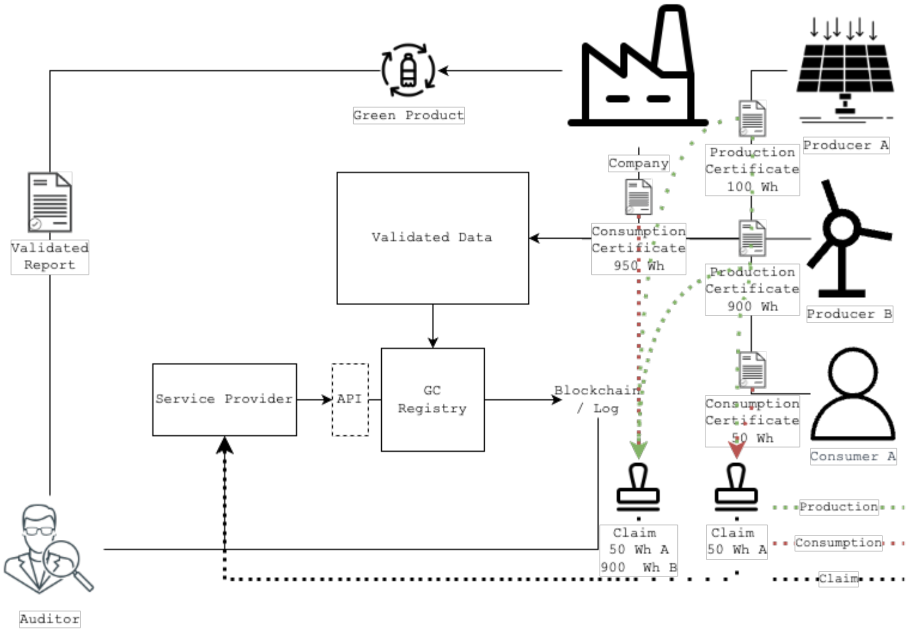

Figure 2 Overview of the system. The registry is deriving validated data and issues time-volume based certificates for production and consumption, that can be claimed directly by consumers or by companies that wants to document the energy going into their products. The claims on the certificates are stored on the registry through a service provider using the registry.

Solution Overview

Below the solution is described with focus on public verifiability. Figure 2 depicts how a registry issues granular certificates to consumers and producers of electricity. For a producer a granular certificate specifies for a given short time interval (e.g., hourly) the amount and specificity of energy produced in that time interval and for consumers the amount of energy used. The registry is trusted to issue this information correctly. While the figure only shows one registry there will be a number of registries.

Everything on the registries happens through the commands listed in Table 1 and depicted in Figure 3. Whenever a certificate is issued or changed then an event corresponding to the state change is logged.

Command	Description
`Issue`	Issues a granular certificate to a consumer or producer.
`Transfer`	Transfers a granular certificate from one consumer or producer to another.
`Claim`	Claims between production and consumption certificates.
`Withdraw`	Withdraws a faulty certificate.
`Remove`	Removes a certificate - when transfers happen between registries.

Table 1 Commands on granular certificates as visualized in Figure 3.

The registry is responsible for maintaining the event log, and it uses a blockchain to ensure that events are immutable and publicly verifiable. In order to ease verification of the entire history of a certificate, each event on the certificate refers to the

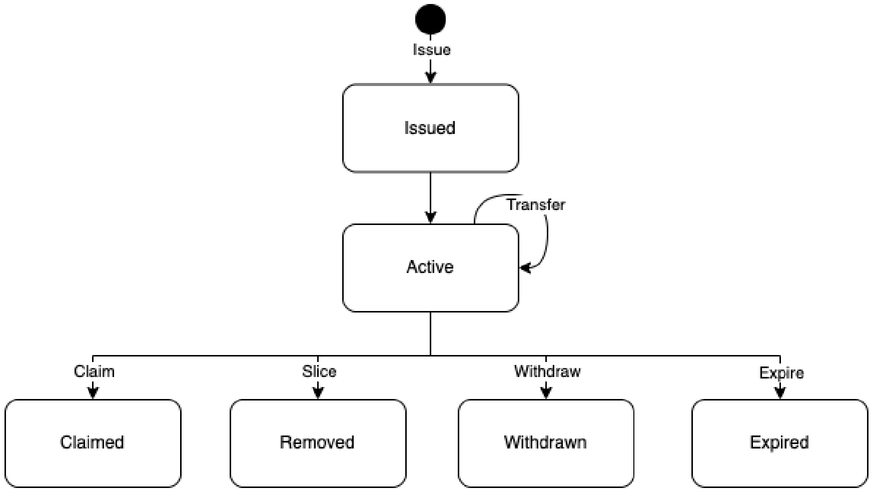

Figure 3 Operations on granular certificates, that reflect the state changes throughout the life-cycle of an issued certificate.

previous event related to that certificate. The ordering defined by the blockchain ensures that the history of each certificate is well defined.

Next, we will describe the most interesting operations on granular certificates, the Transfer and Claim commands, and then the corresponding responsibilities of the registries. The final part of this section deals with public verifiability and auditability of the certificates.

Transfers and Claims

As mentioned a GC contains the entire quantity measured at the meter for the period. In order to enable transferring and claiming parts of a certificate we have borrowed a term from the financial sector, stock-slicing [15], where a single stock can be traded as slices. Similarly, we have introduced slices of a certificate, and a GC therefore consists of two parts:

1 An immutable ”header” which is the collection of attributes on the GC. This data cannot be changed after the GC has been issued. These attributes describe all the properties on the GC, like the grid area, period and which meter the GC originates from. Production certificates further specify the energy source as per the EECS standard [16].

2 A collection of slices. A GC is issued with 1 initial slice corresponding to the total amount of energy. This energy amount may be split in several parts (slices) each with potentially different owners under the constraint that the sum of all slices must equal the original amount of energy in the certificate. An active slice contains two values, the quantity of the slice, and the owner’s public-key. Thus the owner is not identified by a name but by a public key which acts as a pseudonym.

Claims of energy and transfer of ownership happens through the slices and, again, when a slice changes an event is created.

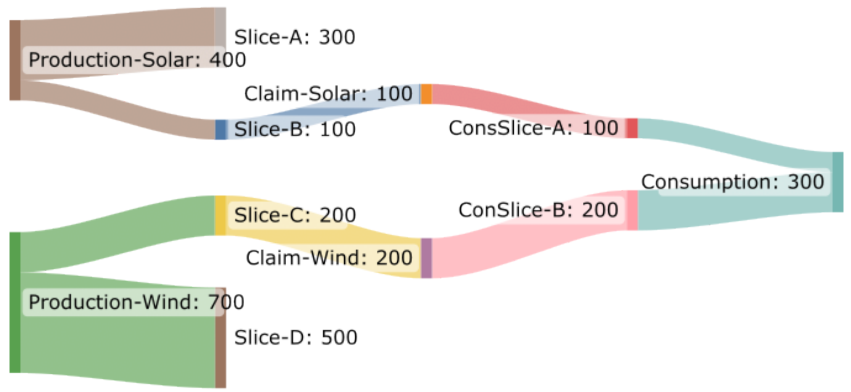

Figure 4 Claims against production from consumption in slices, hence there will be no direct match if slicing is not possible. This feature is essential for matching and transferring of certificates that will enable a marketplace for active certificates across registries.

Transfer of ownership is now carried out by associating the public key of the new owner to a particular slice.

Claiming energy by a consumer is somewhat more involved. The starting point is that a user has a consumption GC and wants to claim the energy produced by a specific source as represented by a production GC (e.g., to prove usage of green energy).

For the sake of example assume that the consumption certificate represents 300 Wh and the production certificate represents 400 Wh and assume that the consumer wants to claim usage of 100 Wh from this producer. The first step is that both consumer and producer split their certificates in two slices.

The consumption certificate is split in slices of 200 Wh and 100 Wh, and the production certificate in slices of 300 Wh and 100 Wh — preserving the totals of each. The 100 Wh consumption slice is then claimed against the 100 Wh production slices, and the latter is marked as used. The remaining 200 Wh in the consumption certificate can then be claimed against another production certificate as illustrated in Figure 4.

Events corresponding to all these state changes are logged by the registry and secured on the blockchain. The section on Confidential Information below describes how this process is carried out while keeping information on energy usage confidential.

Registry

A registry is a single node in the federated network. Each registry can hold any number of GCs and the life-cycle of each GC always stays within the same registry. Figure 5 gives an overview of a registry [17].

The registry is a naive implementation for validation and guarantee of uniqueness of all transactions on its GC's during their lifecycle from issuance, to transfer, claim, removal, withdrawal and expiration of the GCs, see Figure 3. The registry does not

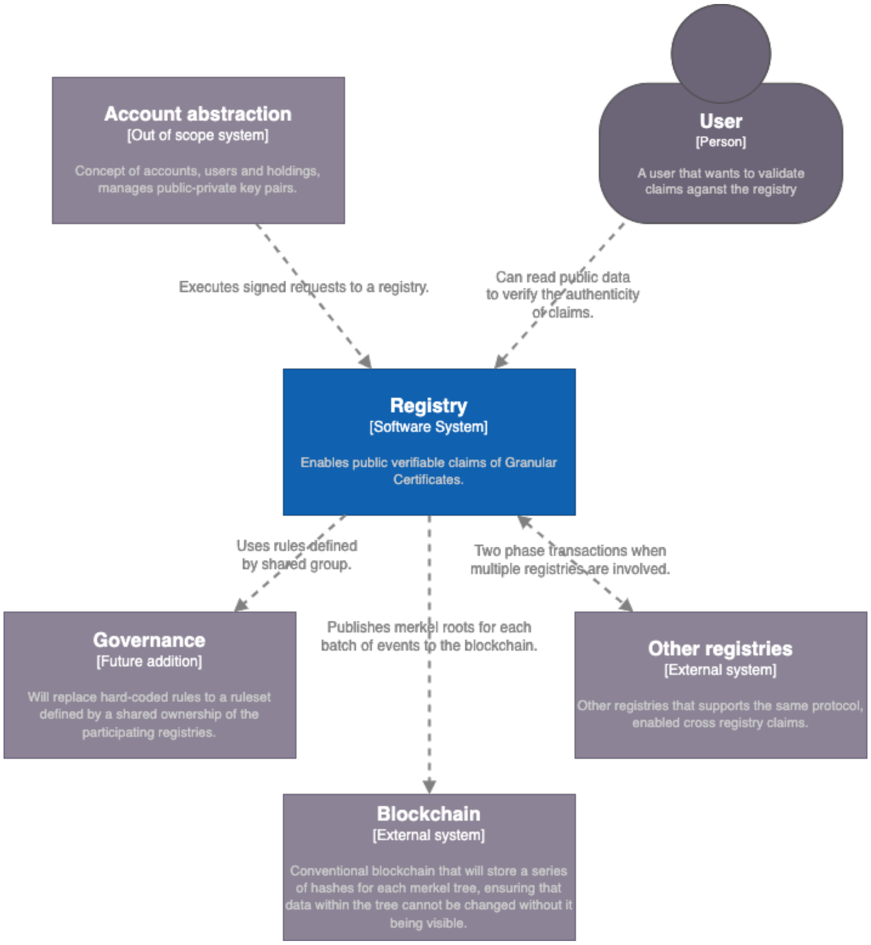
Account abstraction
[Out of scope system]
Concept of accounts, users and holdings, manages public-private key pairs.
User
[Person]
A user that wants to validate claims aganst the registry
Executes signed requests to a registry.
Can read public data to verify the authenticity of claims.
Registry
[Software System]
Enables public verifiable claims of Granular Certificates.
Uses rules defined by shared group.
Two phase transactions when multiple registries are involved.
Publishes merkel roots for each batch of events to the blockchain.
Governance
[Future addition]
Will replace hard-coded rules to a ruleset defined by a shared ownership of the participating registries.
Other registries
[External system]
Other registries that supports the same protocol, enabled cross registry claims.
Blockchain
[External system]
Conventional blockchain that will store a series of hashes for each merkel tree, ensuring that data within the tree cannot be changed without it being visible.

Figure 5 Overview of a registry, it is naive to externalities it merely enables verifiability and uniqueness of proofs. It is a mechanism to enable proofs in a larger system of conventional databases such as event stores, that provides tamper-evidence of issuers and provides uniqueness of claims.

enforce a tamper proof solution, it does however make it possible to validate if tampering has been happening in the past as all certificates and claims can be verified against the event log, which is secured using a blockchain. This approach is justified from the perspective that trust is extended from the TSO's towards the customers and service providers and is acting as a health check and validation mechanism for each registry as they can become faulty due to technical errors, malicious intent or other unknown attack and risk vectors.

For this to work it is essential that users and third parties have the necessary programming interfaces and tools to validate certificates and claims independently of the registry. Each registry is responsible for storing its own event log and must provide an interface enabling users and third party auditors to get the information needed to verify all event logs related to its certificates. The actual data necessary for such validations are described further in the section on Use of Blockchain.

The number of registries can scale with adoption so that a single pricing area and region can have multiple registries and thereby distribute the capacity and workload required for the certification scheme to work reliably at scale.

Note also that a registry must be able to work with other registries. Most importantly, when a consumer belonging to one registry claims energy from a producer belonging to another registry then the corresponding GC slices are in consumption and production certificates issued by the two different registries. Thus, the process of claiming energy involves actions from each registry and we use a two-phase commit protocol in order to ensure that the two certificates end up in consistent states.

Public Verifiability and Audibility

We will now discuss how auditors and other independent parties can audit and verify claims of energy and provably detect conflicting double-spending of energy.

By assumption issuing of a certificate is trusted to be correct (i.e., the TSO is trusted to have measured the right amount of produced/consumed energy and added the correct attributes to the certificate). In order to verify the life cycle of a certificate one must therefore verify that every following change of the certificate is correct according to the event log and verify each event log against the blockchain.

Starting with a production certificate we create a first slice as needed for claims of energy or transfer of parts of the certificate. Then the next slice “consumes” the remainder of the previous slice and so on. So a sequence of updates to a production certificate is valid if the production slices form a chain (as forking corresponds to double spending) and their internal sums equal the amount from the previous certificate that was sliced. Since a certificate refers to the previous version of the certificate a well-defined chain to the original production certificate issued by the registry can be validated.

The same rule applies to the sequence of slicing of a consumer certificate except that forking would mean that the consumer could claim more energy than actually consumed.

Claims of energy by a user are verified by validating that the amount of energy in a particular slice equals the amount of energy in a slice of the production certificate where the energy is claimed from.

Based on the event log and the blockchain the above steps guarantee that every certificate and claim are correctly derived from the initial GC. By validating both the consumption certificate and the production certificate and the claims of energy one is therefore assured that the consumer has used the claimed energy and that it corresponds to the energy from a correctly registered producer.

While the above validation process assures that granular certificates are derived correctly from the previous state, it does not prevent a fraudulent registry deriving two different certificates from a given GC (i.e., forking or double-spending). However, if a registry creates such a fork an immutable proof of this will be created as two different certificates will refer to the same GC. Such a fork can be found by inspecting all certificates created by a registry and following their life cycle in the event log as described above. Alternatively, if two conflicting certificates are seen, by following the life cycle of each back towards the original GC, one will at some point meet a certificate (slice) that was double-spent proving that the registry has misbehaved and the registry creating the conflict can be held accountable.

Use of Blockchain

As discussed above the main objective is to create openness allowing anyone to validate that registries handle energy certificates correctly using a verifiable log that documents all changes.

Blockchain as a Ledger

A blockchain is too inefficient and expensive to be used for storing large amounts of data. Therefore the actual events are stored in databases off-chain in the registry, and a cryptographic hash value (e.g. using SHA256) is registered on the chain. Anyone can then verify the integrity of an event log by validating that the corresponding hash value is registered on the blockchain. In the following, when we talk about registering events on the blockchain, we actually mean that the hash values of these are registered.

By registering each event on a blockchain it is possible to create the desired openness and trust in the event log (and hence the certificates) as the blockchain will guarantee the integrity and ordering of the log entries (cf. [18]). In order to be independent of the trustworthiness of a few servers running the blockchain the best option is to go for a public, permissionless blockchain. While the solution described in this paper is independent of the specific choice of blockchain, the proof-of-concept has been developed using the Concordium blockchain.

As the solution must be able to handle a high number of transactions it was decided not to use smart contracts but rather register data using the native, built-in function which is available on the Concordium blockchain for this purpose. Given the transaction or the block where the data is registered anyone can verify that the data has not been modified.

Another important aspect is to consider the environmental footprint of using a blockchain. Based on the large amounts of energy used by Bitcoin (e.g., on February 15 2023, Cambridge Bitcoin Electricity Consumption Index estimated the annual consumption to be 116 TWh, see [19]), blockchains have gained a reputation of having a very negative environmental footprint. Obviously this does not resonate well in a solution for handling green energy. However, Bitcoin and other early blockchains are based on so-called proof-of-work, which means that each server (mining farm) has to do a lot of computations in order to win the right to build the next block. For each block there is only one winner and the computations of those mining farms not winning are basically wasted. Unlike this, most modern blockchains, including Concordium, are based on proof-of-stake, where the right to create the next block does not require such excessive amounts of computations. It is not possible to give an exact number for the carbon footprint of a public permissionless blockchain as it depends on the number of servers, how these are powered and the load on the blockchain, but studies indicate a carbon footprint of proof-of-stake blockchains corresponding to less than 20 “global persons” (see [20, 21]).

Merkle Trees

While it is cheap and efficient to register data directly on the blockchain, a naive and straightforward registration would fail to handle the expected number of transactions. To illustrate this, a small country such as Denmark has around 3.5 mill. meters, and an hourly certificate for each meter would require almost 1,000 blockchain

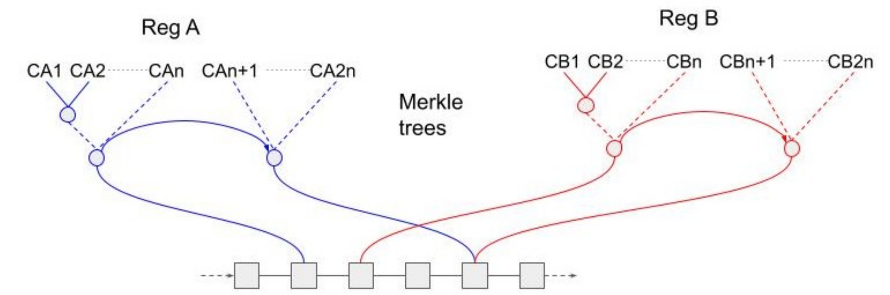

Figure 6 Ordering transaction by posting Merkle tree roots on to the blockchain.

transactions per second for issuing approximately 30 billion certificates yearly, not taking subsequent transactions such as claims on the certificates into account. Clearly, such a solution neither scales to larger countries nor to more fine grained certificates. Another concern is the associated cost of using the blockchain. In a permissionless blockchain the entities running the nodes must be paid for their contribution per transaction. On Concordium, each transaction for registering data costs around 0.01€. Thus the resulting hourly cost of using the blockchain would be 35,000€ or 840,000€ per day for Denmark only, which is neither sustainable nor worthwhile for any operator or service provider.

A key point to solve this scalability problem is to observe that it is not important that log entries are registered immediately on the blockchain, so there is no requirement for immediate consistency as in financial transactions. The solution can tolerate a certain latency as long as the certificates are publicly verifiable with eventual consistency. We will get back to how the latency can be managed.

Rather than registering each event immediately on the blockchain each registry locally batches its registrations in a Merkle tree [22], see also Appendix Merkle Trees. When the Merkle tree is full the registry commits all registrations in the tree by registering the root of the Merkle tree in a single blockchain transaction, and the registry then starts building a new Merkle tree for the subsequent transactions.

This gives a total ordering of all transactions from a registry as defined by the ordering of the Merkle trees on the blockchain and the ordering of registrations within each Merkle tree as illustrated in Figure 6.

A log entry now consists of the following information

1 The actual information in the event. This consists of the public certificate data, which is shared and can be verified on the blockchain, and private data belonging to the owner of the certificate (the section below on Confidential Information describes how the latter is handled and used).

2 A proof that the hash value of the event is included in a particular Merkle tree. This proof includes a list of hash values (corresponding to the height of the tree — logarithmic in the size of the tree) enabling verification that the hash value of the given event is indeed included in the Merkle tree with the given root, see Appendix Merkle Trees.

3 The transaction or block in which the root (a hash value) of the Merkle tree is indeed registered on the blockchain.

A receiver of a granular certificate or claim (the public data) can now validate that the this was created at a particular time by a particular registry by getting the corresponding event (or events) and validating that

- each event is included in a Merkle tree (using the proof in item 2 above) and that each Merkle tree is registered on the blockchain (using the value in item 3 above); and
- based on the ordering defined by the registrations the event(s) document the state of the certificate or claim.

The first step can and must be done using generic tools for inspecting the blockchain in order to make this verification independent of the registry. Furthermore, the validation of inclusion in the Merkle tree is done without knowing other events in the Merkle tree. Only, the sequence of hash values along the path in the tree from the hash value of the given entry to the root is needed, see Appendix Merkle Trees.

Balancing scalability and latency. Batching registrations in a Merkle tree allows the registry to balance scalability and latency as it is very fast, efficient and cheap to add a new registration to a Merkle tree compared to making a registration on a blockchain. Therefore, to optimize throughput and minimize costs a registry would tend to build large Merkle trees. As the validation of inclusion in the tree only depends logarithmically on the size of the tree, validation remains efficient for large trees. However, the downside of this is that a certificate is not openly committed before the root of the Merkle tree is registered on the blockchain. Or in other words, the larger the Merkle tree, the longer the latency until the certificate is openly committed and eventually consistency is achieved.

By adjusting the size of the Merkle tree the registry can therefore balance scalability and cost against latency. Initially, we expect to use Merkle trees of height 10-20 corresponding to 1,000-1,000,000 data entries in each tree, but this choice will be adjusted along the way as an optimization parameter. The effect on cost is immediate, as even with small trees containing 1,000 entries the daily blockchain costs drops from the 840,000€ mentioned above to 840€.

Local Merkle trees for each registry. The solution builds up local Merkle trees for each registry. One could also have designed the solution with a general layer 2, which batches transactions from all registries in a Merkle tree and commits the Merkle tree root to the blockchain, when it is full.

However, the solution with local Merkle trees, that we opted for, allows each registry to balance latency independently of other registries and, perhaps more importantly, anybody can see on the layer 1 blockchain who registered a given Merkle tree. This makes it quite easy to locate all registrations from a particular registry.

Confidential Information

The ownership information and energy amounts in granular certificates are confidential and may not be exposed in the public event log. In this section, we show how to protect this information while ensuring that a malicious actor cannot violate validity.

Storing Energy Information

We use Pedersen commitments [23] to hide the energy amounts of certificates and slices in the event log. To ensure validity, these logs must contain proofs of correctness which we specify in the following. We note that the owner of a certificate/slice can still reveal information about the amounts, as they know the commitment opening information. For more information on Pedersen commitments see Appendix A.

Proof of correctness for slices. The following proof of correctness is used both for slicing production and consumption certificates. The proof is described for a single slice, but recall that a certificate may be sliced several times. Each slice contains three amounts of energy; namely the total amount, the claimed amount, and the remainder. A correct slice satisfies the following two properties:

- The amounts of energy add up correctly, that is total energy equals the sum of claimed and remaining energy — thereby avoiding double counting.
- Each amount of energy is in a specific range; in particular amounts are non-negative.

The total amount corresponds to the remainder of the previous slice or if this is the first slice to the initial amount in the credential.

Denote by $\mathsf{C}_{\mathsf{total}}, \mathsf{C}_{\mathsf{claim}}, \mathsf{C}_{\mathsf{remain}}$ the commitments to total, claimed, and remaining energy amounts. In the proof of concept the correctness properties are proved as follows.

The sum property is proved using a sigma protocol. First, using the homomorphic property of Pedersen commitments commitment $C = \mathsf{C}_{\mathsf{claim}} \cdot \mathsf{C}_{\mathsf{remain}} \cdot \mathsf{C}_{\mathsf{total}}^{-1}$ is computed. For the sum property to hold C must be a commitment to 0 which for Pedersen commitment has the form $g^0 h^r = h^r$ where (g, h) is the commitment key and r the opening information. The sum proof reduces to proving the statement (proof of knowledge of discrete logarithm):

I know r such that C has the form h^r.

This type of statement can be proven using a Sigma protocol for the discrete logarithm relation (see [24] for an overview on the topic). The actual proof is non-interactive (via the Fiat-Shamir transform [25]).

An alternative approach is to show the sum property indirectly. Here $\mathsf{C}_{\mathsf{remain}}$ is defined as $\mathsf{C}_{\mathsf{total}} \cdot \mathsf{C}_{\mathsf{claim}}^{-1}$ In particular, $\mathsf{C}_{\mathsf{remain}}$ is not actually stored in the certificate itself, but computed from the other two commitments whenever needed. This ensures that the sum property holds by definition.

The range property is shown using Bulletproofs [26] which allows to prove that a committed value is in a range of the form $[0, 2^k]$, e.g. $k = 20$. The proofs for all energy amounts in the certificate can be aggregated to save space. The proofs are again used in the non-interactive form. Implementation: For commitments and range proofs the Dalek bulletproof library is used [27]. The sigma protocol is implemented on top.

Finally, for consistency between slices, one can simply use the previous $\mathsf{C}_{\mathsf{remain}}$ as the $\mathsf{C}_{\mathsf{total}}$ in the current slice. If for some reason a fresh commitment is required, the proof used in the proof of correctness for matching described below can be used.

Proof of correctness for matching. When a consumer claims a certain amount of produced energy, one needs to ensure that the claimed amount is the same in both the production and the consumption slice.

As a building block, we use the following Sigma protocol to show that two given commitments contain the same value. Let C_1 and C_2 be given commitments to message m where the prover knows the opening information for both. Let $C = C_1 \cdot C_2^{-1}$. If the commitments contain the same value, C must be a commitment to 0, i.e., it has the form $g^0 h^r = h^r$ where (g, h) is the commitment key and r the opening information (derived from the opening information of the given commitments). So using the discrete logarithm proof from the previous section, we can show that C is a zero commitment and thus both commitments contain the same value.

Normally, the production and consumption slices have different owners. This means we cannot create a proof matching their claim commitments directly as this would require knowing both opening information. Instead we can use the following protocol to create a matching proof between claim commitments $\mathsf{C}_{\mathsf{prod}}$ and $\mathsf{C}_{\mathsf{cons}}$

One owner (e.g. of the consumption slice) creates a fresh commitment $\mathsf{C}_{\mathsf{mid}}$ to the claim value with opening information r_{mid} as well as a proof that $\mathsf{C}_{\mathsf{mid}}$ contains the same value as their own claim commitment. The owner sends $\mathsf{C}_{\mathsf{mid}}$, r_{mid}, and the proof to the other owner (here of the production slice). The other owner now creates a proof that their claim commitment contains the same values as $\mathsf{C}_{\mathsf{mid}}$. The two proofs and $\mathsf{C}_{\mathsf{mid}}$ now form a proof of correctness for matching the claimed production and claimed consumption. The value r_{mid} can be safely deleted.

Ownership Information

The owner of a certificate is represented by a public key to a signature scheme, i.e., whoever knows the secret key “owns” the certificate. To avoid public linkability, owners need to generate a fresh key for each new certificate. This is not much different from UTXOs e.g. on Bitcoin. The core issue in both cases is that the owner now needs to store many keys. A common approach to mitigate this issue is the use of a key derivation scheme where all keys are computed from a master secret. This approach is also known as a (hierarchical) deterministic wallet where the master secret is often represented as a seed phrase. For more technical details see for example [28].

Conclusion and Future Work

In this paper we have shown how blockchain technology and well-known cryptographic techniques such as commitments, zero-knowledge proofs and Merkle trees, can be used to create a solution for tracking and verifying very high volumes of transactions. This bridges the gap from conventional centralized big-data systems towards decentralized structures that leverage trust and credibility.

While the concepts can be used for many other applications, a first version of the solution has been built for handling electricity certificates in order to increase the trust in such certificates and prevent greenwashing. Electricity certificates include confidential data and we have shown how to protect the privacy of such information while ensuring consistency. Furthermore, the solution ensures that the certificates and claims can be independently verified against the event log and conflicting certificates will leave an immutable trace in the solution.

This first version has been built to show that a decentralized, scalable solution using blockchain is indeed possible. This version is considered the start of a journey with a number of possible improvements for scalability, consistency checks and auditability. The solution would, in particular, be strongly improved by having a set of independently developed tools for validating the life cycle of GCs.

Another possible improvement which enhances decentralization would be to register consumption and registration on the blockchain much earlier in the process — in the extreme case at the meters. This would reduce the assumed trust on TSOs to issue correct consumption and production certificates if the energy amounts in these certificates can be verified against measurements closer to the consumer and producer. The latter could allow for very detailed tracking of energy and add further credibility to claims.

Finally, another topic for future work is related to the way event logs are verified on the blockchain. In the solution presented in this paper, validation of a claim or certificate involves that one must look through all related event logs. If all these related logs where collected together on the blockchain (e.g., using a smart contracts) it would conceivably be much easier to validate a the lifecycle of a certificate and it could potentially prevent double-spending of energy. Providing such a low-cost and scalable solution for this would be very interesting.

Terminology

Batch A batch of transactions that are hashed using Merkle trees and published to an external blockchain

Claim A claim between production and consumption GC's matching a slice of the consumption certificate with a slice of the production certificate. A unique proof of origin of the underlying asset.

Event Log A log containing the transactions on a Registry

Expire A GC has a lifecycle and it expires at a set date. The expiration is a governing policy issue.

Granular Certificate (GC) A time based certificate for production and consumption of energy.

Issue The process of creating a GC in accordance with validated data.

Proof-of-Stake (PoS) The right to create the next block in the blockchain is assigned to a node based on the (locked) stake of the nodes.

Proof-of-Work (PoW) The right to create the next block in the blockchain is assigned to the node that first solves a difficult problem.

Registry A data structure where GCs are stored and transactions committed in an eventual,consistent manner.

Slice A part of the energy in a certificate. Used to transfer parts of a production certificate and to match consumption claims against production.

Transfer Transfer ownership of whole or partial amount of energy in a GC.

Unspent transaction output (UTXO) A transaction mechanism used to handle crypto currency on some blockchains. If a user only spends a part of the amount received in a transaction, the remaining, unspent part goes back to the user as an UTXO and can be used in subsequent payments.

Withdraw In order for the issuer to recall or withdraw a GC if the validated data changes for some reason. Quality assurance mechanism.

Availability of data and materials

The source code of this project can be found at [1].

Author's contributions

The authors are sorted alphabetically.

Morten Jokumsen Solution Overview and implementation of proof of concept

Torben Pryds Pedersen Use of Blockchain and Confidential Information

Martin Schmidt Daugaard Solution Overview, requirement clarification and implementation of proof of concept
Daniel Tschudi Use of Blockchain and Confidential Information
Mikkel Wienberg Madsen Confidential Information and implementation of proof of concept
Thomas Wisbech Solution Overview and requirement clarification

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Mjølner Informatics,mjolner.dk. [2]Concordium,concordium.com. [3]Energinet,energinet.dk. [4]Alexandra Instituttet,alexandra.dk.

References

1. Energinet: Project Origin. https://github.com/project-origin/registry
2. Track, E., Trace: Energy Track and Trace. https://energytrackandtrace.com/
3. EnergyTag: EnergyTag Initiative. https://energytag.org/
4. EnergyTag: Granular Certificate Scheme Standard. Technical report, EnergyTag Initiative, London, UK (March 2022). https://energytag.org/wp-content/uploads/2022/03/20220331-EnergyTag-GC-Scheme-Standard-v1-FINAL.pdf
5. United Nations' High-Level Expert Group on the Net Zero Emissions Commitments of Non-State Entities: Integrity Matters: Net Zero Commitments by Businesses, Financial Institutions, Cities and Regions. Technical report, United Nations (November 2022). https://www.un.org/sites/un2.un.org/files/high-levelexpertgroupupdate7.pdf
6. Babel, M., Gramlich, V., Körner, M., Sedlmeir, J., Strüker, J., Zwede, T.: Enabling end-to-end digital carbon emission tracing with shielded nfts. Energy Informatics **5** (2022)
7. Kai: Truly green energy trades with blockchain-based certificates: A proof of concept to embed a certification system in Elia Group's consumer centric market design (2022)
8. Schmidt, M.: PROJECT ORIGIN – How can we guarantee origin of electricity? www.linkedin.com (2019)
9. Energinet: Energy Origin. https://en.energinet.dk/energy-data/datahub/energy-origin/
10. Louw-Reimer, J., Nielsen, J.L.M., Bjørn-Andersen, N., Kouwenhoven, N.: In: Lind, M., Michaelides, M., Ward, R., Watson, R.T. (eds.) Boosting the Effectiveness of Containerised Supply Chains: A Case Study of TradeLens, pp. 95–115. Springer, Cham (2021)
11. Maersk: A.P. Moller - Maersk and IBM to discontinue TradeLens, a blockchain-enabled global trade platform (2023)
12. München, F.: Das Projekt InDEED - FfE (2023). https://www.ffe.de/projekte/indeed/
13. Buldas, A., Draheim, D., Gault, M., Laanoja, R., Nagumo, T., Saarepera, M., Shah, S.A., Simm, J., Steiner, J., Tammet, T., Truu, A.: An ultra-scalable blockchain platform for universal asset tokenization: Design and implementation. IEEE Access **10**, 77284–77322 (2022). doi:10.1109/ACCESS.2022.3192837
14. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains, pp. 17–30 (2016). doi:10.1145/2976749.2978389
15. of Investor Education, S.O., Advocacy]: Fractional Share Investing - Buying a Slice Instead of the Whole Shares. Technical Report 2, US Securities and Exchange Commission (February 2021). https://www.sec.gov/oiea/investor-alerts-and-bulletins/fractional-share-investing-buying-slice-instead-whole-share
16. Moody], P.: EECS Rules Fact Sheet 5 Types of Energy Inputs and Technologies. Technical Report 5, Association of Issuing Bodies (December 2019). https://www.aib-net.org/eecs/fact-sheets
17. Energinet: Project Origin - Verifiable Eventstore. https://github.com/project-origin/registry/tree/main/doc/architecture/verifiable_event_store
18. Wüst, K., Gervais, A.: Do you need a Blockchain? Cryptology ePrint Archive, Report 2017/375. https://eprint.iacr.org/2017/375 (2017)
19. for Alternative Finance, C.C.: Cambridge Bitcoin Electricity Consumption Index. https://ccaf.io/cbeci/index
20. PricewaterhouseCoopers Advisory: Study of the environmental impact of the Tezos blockchain Life Cycle Assessment of the Tezos blockchain protocol. Technical report, Nomadic Labs (December 2021)
21. Carbon Crowd: Internet Computer Footprint: Assessing IC Energy Consumption and Sustainability. The Internet Computer Review (2022)
22. Merkle, R.C.: A digital signature based on a conventional encryption function. In: Pomerance, C. (ed.) CRYPTO'87. LNCS, vol. 293, pp. 369–378. Springer, ??? (1988)
23. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 129–140. Springer, ??? (1992)
24. Ivan Damgård: On Σ-Protocols. https://www.cs.au.dk/~ivan/Sigma.pdf (2010)
25. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, ??? (1987)
26. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE Symposium on Security and Privacy, pp. 315–334. IEEE Computer Society Press, ??? (2018)
27. Henry de Valence, Cathie Yun, Oleg Andreev: Dalek Bulletproofs. https://github.com/dalek-cryptography/bulletproofs (2018)
28. Marek Palatinus, Pavol Rusnak, Aaron Voisine, Sean Bowe: Mnemonic code for generating deterministic keys. Technical Report 39, Bitcoin (September 2013). https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

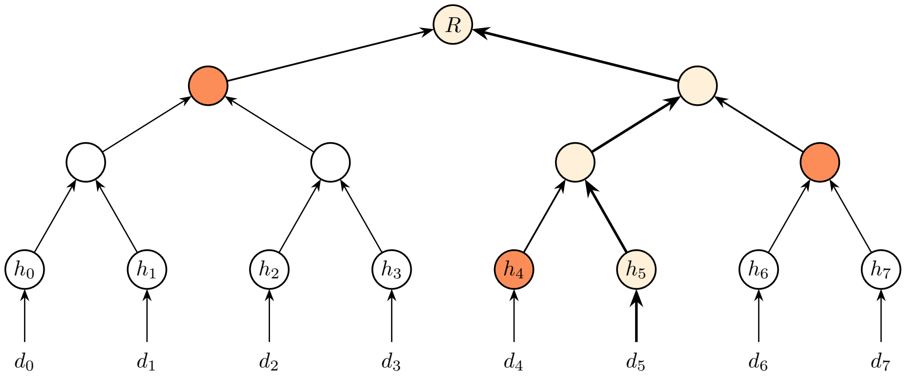

Figure 7 Merkle tree with 8 elements. The hash values in the dark nodes are needed to validate that data element d_5 is included in the root, R.

Appendix A: Pedersen Commitments

A commitment scheme allows a party to commit to a value v without revealing the value. Later the party can reveal the value by opening the commitment.
Let $\mathbb{G}$ be a group of prime order q with generators g, h.

Definition 1 ([23]) *The Pedersen commitment scheme consists of the following functions:*

- *The* commit *function* Com *which takes as input the value v and randomness r. It outputs* $(\mathsf{C}, o) := (g^v h^r, r) \leftarrow \mathsf{Com}(v; r)$ *where* C *is the* commitment *and o the* opening information.
- *The* opening verification *predicate* VerifOpen *which takes as input the value v, the opening information o, and commitment* C. *It returns true if and only if* $g^v h^o = \mathsf{C}$.

The Pedersen commitment has the following two properties:

Perfectly Hiding The commitment $\mathsf{C} \leftarrow \mathsf{Com}(v; r)$ is statistically independent of the value v.

Binding Assuming that DDH assumption is hard in $\mathbb{G}$ and the pair-wise discrete log between g and h is unknown, it is computationally hard to find o', v' such that for given commitment $\mathsf{C} \leftarrow \mathsf{Com}(v; r)$ it holds that $\mathsf{VerifOpen}(v', o', \mathsf{C}) = 1$.

Appendix B: Merkle Trees

A Merkle tree provides a structured way to compute a hash value of a number of data elements, so that one can efficiently verify that a particular data value is included without knowing the remaining data values.
As the name suggests this is based on a hashing the data items in a tree structure, where the root of the tree represents the resulting hash value and hash values of the data values are in the leaves. Figure 7 illustrates a Merkle tree with 8 data elements. Each node represents the hash value computed from inputs as indicated by the arrows. The root, R, represents the hash value of $d_0, d_1, ...d_7$. To validate that data element d_5 is included in R, the hash values in the three dark nodes are needed in order to validate the path from h_5 to R.