ALLIANDER

Decentralized authorization of access to energy data

Erwin Rooijakkers
September 26, 2017

1 INTRODUCTION

Within Alliander various projects (like Tippiq, HelloData, and GateKeeper) work on the use case of authorizing the access of energy data for P1-device to app, so that the consumer is in control over his data. Because of the characteristics of the blockchain (immutability, decentralization, independence, automatic settlement, availability, reachability), it seems beneficial to decentralize the authorization business logic on a public or private blockchain. In this document a prototype for authorizing data access on the Ethereum blockchain is introduced, and future recommendations are provided.

The Ethereum blockchain is an immutable, distributed ledger with a virtual machine [1]. Using peer-to-peer technology, cryptography, and consensus formation algorithms, no trusted third party is necessary to keep an "official" copy of the state of the Ethereum Virtual Machine (EVM). Code in so-called "smart contracts" can be executed and publicly verified. To change the state of the blockchain fees have to be paid in the cryptocurrency (Ether). Querying the Ethereum blockchain is free and can be done by anyone with access to the ledger.

In Figure 1.1 obtained from [1] based on [2] the Ethereum blockchain is visualized. The Ethereum blockchain is a “chain” of “blocks” that is shared over every full node in the network. Each block consists of a hash of the previous block, transactions (transferring the cryptocurrency Ether from one address to another), receipts (state changes) and state (the state of the EVM). Because the Ethereum blockchain contains state alongside transactions, on the EVM a Turing complete programming language is available that enables smart contracts [1].

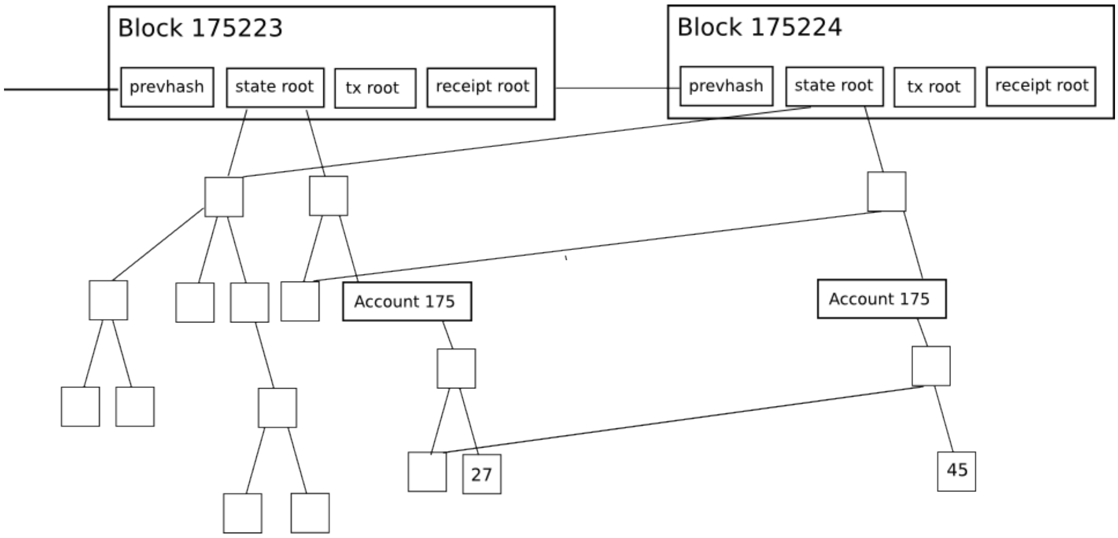

Figure 1.1: Ethereum blockchain stores three kinds of information: transactions, receipts, state.

To reach consensus on the state of the "real" blockchain (to guard against "double spending" attacks where an attacker makes a longer official chain) various algorithms can be used; the most famous ones being Proof of Work (PoW) and Proof of Stake (PoS). In PoW consensus is reached by the solution of a cryptographic puzzle over the state and transactions, that takes computer power to solve. Because it takes so much computer power to find a solution, the network can trust that the result was reached by consensus (it would take an attacker at least 51% of computer power to built a longer chain). In PoS trust is reached because people who are invested in the network set aside their cryptocurrency to stake for receipts and transactions. Ethereum currently works with PoW, but will switch to PoS sometime this year [1].

Now, authorizing energy data on the Ethereum blockchain, and its benefits and limitations, will be discussed.

2 Authorizing energy data on the blockchain

There are three entities at play when authorizing access to energy data:

1. Device (IoT-device that provides energy data)
2. Consumer (owner of an IoT-device)
3. App (service provider that wants to do something with energy data)

The consumer, app, and device need to have an address on the Ethereum blockchain (key pair). Because of the properties of blockchain addresses it can be ensured that the actions are executed by the entities involved. A device can be claimed by a consumer. And a consumer can then give access to the app. Finally, a device can check if an app has access.

A smart contract that implements this logic is shown below in 3 Prototype.

3 Prototype

```
pragma solidity ^0.4.10;


import "Mortal.sol";


contract SmartEnergyAuthorizations is Mortal {

  // Mapping from device to consumer
  mapping(address => address) claims;

  // Mapping from device to app to authorization flag
  mapping(address => mapping(address => bool)) authorizations;

  // Constructor
  function SmartEnergyAuthorizations() {
    owner = msg.sender;
  }

  function claimDevice(address consumer) {
    address device = msg.sender;
    claims[device] = consumer;
  }

  function authorize(address device, address app) {
    address consumer = msg.sender;

    require(claims[device] == consumer);

    authorizations[device][app] = true;
  }

  function revoke(address device, address app) {
    address consumer = msg.sender;

    require(claims[device] == consumer);

    authorizations[device][app] = false;
  }

  function isAuthorized(address app) constant returns (bool) {
    address device = msg.sender;

    // NOTE: Returns false when no entry for device, or for
    // device and app, exists in the `authorizations` mapping.
    return authorizations[device][app];
  }

}
```

The above smart contract is deployed on the Ethereum test net. It is possible to interact with this smart contract back-end via the front-end deployed on `http://erooijak.simple-webhosting.eu`.

4 Discussion

There are some limitations to authorizing access to energy data on the Ethereum blockchain and in the smart contract above. These are stated below as problem, and underneath every problem a possible solution is provided.

- **Problem:** Performing the transactions (e.g., claiming a device) costs money (fees on Ethereum blockchain). This means that the consumer and device have to pay.
  Solution: The IOTA Tangle blockchain does not have this limitation [3]. It can, in the future, also share energy data between IoT-devices in an end-to-end encrypted manner [4] (!).

- **Problem:** Scalability of the blockchain. When more devices start using the Ethereum blockchain state changes and transactions might not be included in blocks. This can lead to increased transaction fees or longer during transactions.
  Solution: The IOTA Tangle [3] is a "blockchainless blockchain" where consensus formation is not separated from the performing of transactions (every previous transaction needs to validate two earlier transactions). This is why the IOTA Tangle has no scalability issues. It would solve this problem. (Note: IOTA Tangle does not yet implement smart contracts because timestamps are difficult to implement, so another way to do authorization needs to be found, or wait till the problem is solved, which it is almost, e.g., [5]).

- **Problem:** Scalability of implementation. If a lot of devices and authorizations are stored in the smart contract it might lead to performance problems when it stores many devices and authorizations.
  Solution: Use an appropriate data structure; or use offchain or off-Tangle processing.

- **Problem:** In the current smart contract, when a device changes ownership the old authorizations are taken along to the new consumer. The reason is that a mapping type does not know which keys it contains.
  Solution: To fix this another data structure needs to be used, that can be iterated over and cleaned when switching ownership. For example an IterableMapping.

- **Problem:** In the current smart contract, the list of devices for a consumer are not iterable. To provide a convenient user interface around the smart contract back-end this type of information needs to be queryable. But maybe such a UI is not a requirement.
  Solution: Same as above.

- **Problem:** A consumer cannot provide a granularity of access to a device (e.g., only part of the data of a device, like only realtime data but not the history).
  Solution: To fix this other data structures need to be used in the contract, like the

IterableMapping mentioned above. Wherein also the capabilities of the device can be stored.

- **Problem:** Authorization data is publicly available and inspectable.
  Solution: This can be solved by a proper anonymous identity management system like IRMA [6] or Sovrin [7], where only the attributes of an identity that are necessary (e.g., does a consumer live on the address of the device?) are available, in a non-identifiable manner.

5 Conclusion

Authorizing access to energy data on the Ethereum blockchain is doable. It makes all the benefits of the blockchain automatically available. The authorization data information cannot be tempered with, it provides an immutable audit trail of events, and it is convenient for a device to access the necessary authorization information, or even for an app to access the data of a device, even when the device is behind a firewall.

There are some limitations to using a blockchain for authorizing access to energy data. Limitations coming from high transaction costs, the data structures used in the contract, and identity management. A combination of IOTA Tangle, new data structures, and an identity management system (like IRMA [6] or Sovrin [7]) will help solve this.

I am convinced of the benefits of blockchain technology for authorizing (and transporting) energy data and hope that we will build a data authorization gateway backed by blockchain technology.

References

[1] V. Buterin. A next-generation smart contract and decentralized application platform. `https://github.com/ethereum/wiki/wiki/White-Paper`. Accessed 09-18-2017.

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `https://bitcoin.org/bitcoin.pdf`. Accessed 09-18-2017.

[3] S. Popov. The tangle (version 0.6). `https://iota.org/IOTA_Whitepaper.pdf`. Accessed 09-18-2017.

[4] D Sønstebø. Iota development roadmap. `https://blog.iota.org/iota-development-roadmap-74741f37ed01`. Accessed 09-26-2017.

[5] S. Popov. On the timestamps in the tangle. `http://iota.org/timestamps.pdf`. Accessed 09-26-2017.

[6] B. Jacobs. I reveal my attributes. https://www.irmacard.org/irma/. Accessed 09-18-2017.

[7] Sovrin Board of Trustees. Sovrin provisional trust framework. https://sovrin.org/wp-content/uploads/2017/06/SovrinProvisionalTrustFramework2017-03-22.pdf. Accessed 09-18-2017.