THE JBBA

PEER-REVIEWED RESEARCH

OPEN ACCESS
ISSN Online: 2516-3957
ISSN Print: 2516-3949
https://doi.org/10.31585/jbba-1-2-(6)2018

Competing Interests:
None declared.

Ethical approval:
Not applicable.

Author's contribution:
PR[1] designed and coordinated this research and prepared the manuscript in entirety.

Funding:
None declared.

Acknowledgements:
PR[1] acknowledges Dr Catherine Jones, Dr Marius Portmann, Dr David Hyland-Wood and Dr Shahan Khatchadourian for their feedback and suggestions on this paper.

Using Ethereum Registration Authorities to establish Trust for Ethereum Private Sidechains

Peter Robinson[1]

Protocol Engineering Group and Systems (PegaSys), ConsenSys
School of Information Technology and Electrical Engineering, University of Queensland, Australia
Correspondence: peter.robinson@uqconnect.edu.au
Received: 11 September 2018 **Accepted**: 16 October 2018 **Published**: 22 October 2018

Abstract

Ethereum Private Sidechains are permissioned Ethereum blockchains which allow authorised participants to interact privately using Smart Contracts. Permissioned blockchains are appropriate for use in scenarios in which the list of blockchain participants and the code and state of contracts on the blockchain must be kept secret. Ethereum Registration Authorities are a system of Smart Contracts which can be used to resolve bootstrap information based on domain names to allow Ethereum Private Sidechains to be established between parties which have not previously interacted. This paper presents the architecture, design, and gas usage of a reference implementation for the Ethereum Registration Authority system. It analyses the security properties of the system and shows that it is secure, decentralized, and censorship resistant. The reference implementation gas usage is analysed and shown to be independent of the length of domain name and number of entries in the Smart Contracts.

Keywords: *Ethereum Registration Authority, Ethereum Private Sidechains, enterprise, permissioned, private, blockchain, era*

JEL Classifications: *D02, D71, H11, P16, P48, P50*

1. Introduction

Ethereum Private Sidechains [1] are intended to be private Ethereum blockchains which can be established on-demand between previously unrelated organisations in a similar way to how a user of a web browser can establish a secure connection to a web server by simply entering in a URL such as https://example.com/. They will allow only permissioned nodes belonging to participating organisations to join the sidechain's peer-to-peer network and only allow permissioned accounts belonging to participating organisations to submit transactions to the nodes. These sidechains will facilitate a new wave of blockchain adoption by enabling complex new use-cases such as supply chains which include confidential business logic. They will provide the privacy and permissioning required by enterprises [2].

The strong privacy and permissioning features required by enterprises are in contrast to the permissionless Ethereum MainNet [3] which provides strong non-repudiation properties with no privacy or confidentiality [4]. However, these strong non-repudiation properties mean that Ethereum MainNet is the ideal location for securely storing data for which the authenticity and integrity of the data is paramount.

To establish an Ethereum Private Sidechain, the node IP addresses, verification public keys, encryption public keys, and accounts belonging to each organisation needs to be determined. This linking of real-world identity to blockchain information needs to be done securely. This article describes and analyses Ethereum Registration Authorities, an Ethereum smart contract-based approach for establishing bootstrapping information for Ethereum Private Sidechains.

This paper reviews existing techniques for establishing trust, showing the existing techniques to be not appropriate for Ethereum Private Sidechains. It then describes the architecture of Ethereum Registration Authorities. The design of the system is discussed, explaining how the system minimizes complexity and gas usage, reduces round trips caused by multiple calls, whilst being extensible and providing organisations with autonomy. The gas usage of the Ethereum Registration Authority reference implementation [5] as measured using the

Truffle test framework in combination with the Truffle EVM emulation is presented. The paper closes with a discussion of the security of Ethereum Registration Authorities.

2. Existing Techniques

A range of techniques have been devised to establish private blockchain networks. JP Morgan's Quorum [6], a fork of Go Ethereum [7], requires network addresses and keys to be shared out-of-band.

Similarly, Hyperledger Fabric requires the IP addresses of "anchor peers" and root CA certificates for each participating organisation to be distributed out-of-band to establish a network [8]. Requiring this out-of-band sharing between parties precludes the establishment of networks on demand between previously unrelated organisations and as such is not a suitable technique for establishing trust for Ethereum Private Sidechains.

R3's Corda [9] uses a centralised "Network Map Server" to distribute bootstrapping certificates which are ultimately signed by a single root "Network Certificate Authority". Having these points of architectural and political centralization goes against the core tenants of Ethereum [10], and the Internet [11] around decentralization. Though Ethereum Private Sidechains may be configured such that a certain party is explicitly given greater control than others, perhaps because this is required for regulatory compliance reasons, the system should be able to operate in a decentralized manner to allow it to be applicable in the greatest number of use-cases. As such, having a centralization point similar to Corda's "Network Map Server" is not an appropriate technique for establishing trust for Ethereum Private Sidechains.

InterPlanetary File System (IPFS)'s InterPlanetary Name System (IPNS) and Monero use DNS TXT records to store bootstrap information [12] [13]. Domain Name Registrars typically exercise due diligence when issuing domain names such that they ensure that the domain names legitimately relate to the organisation's real-world identity. As such, if the information in DNS could be trusted, it would provide a mechanism for determining bootstrap information for previously unrelated organisations. However, DNSSEC, the additions to DNS to make it secure, are not widely deployed around the globe [14] [15]. This means that using DNS TXT records is not an effective method of securely deploying bootstrap information. Additionally, domain name registrars control the issuance and resolution of domain names under their control. This results in political centralization and can be used for censorship. As such, using DNT TXT records is not an appropriate method to establish trust for Ethereum Private Sidechains.

NameCoin [16] proposed a decentralized name-value pair system based on Bitcoin to be used as a decentralized DNS. Using this system for the storage of Ethereum Private Sidechain bootstrap information would require applications to have both a NameCoin client and an Ethereum client, which means introducing more software, which increases the maintenance cost and complexity of the system. Additionally, this system is no longer secure as the majority of the mining of NameCoin is now done by one organisation [17]. This organisation could re-write the history of the NameCoin blockchain given it mines more than 51% of the currency.

EthDNS [18] has been proposed as an approach to storing DNS information in Ethereum Name Service (ENS) [19]. The authors espouse not having a hierarchy of name servers by storing all domains in the one contract. This lack of hierarchy is motivated by the desire to be immune from censorship by name servers delisting intermediate domains. However, having all domains listed in one contract provides no mechanism to distinguish highly trusted domains from less trusted domains. For example, it could be imagined that some registrars might complete significant audits prior to listing a domain, whereas other registrars may require minimal information. Additionally, EthDNS does not provide a way for organisations to certify a list of domains which should be used for a certain purpose. For example, with EthDNS, there is no way for a government regulator to specify which organisations are approved to offer a certain service. Ethereum Private Sidechains need to be able to group domains according to different types of trust to enable to maximum number of use-cases. As such, EthDNS is not an appropriate technique for establishing trust for Ethereum Private Sidechains.

This article introduces the Ethereum Registration Authority architecture, explaining how this architecture overcomes the deficiencies of previous techniques. In particular, this technique allows for sharing of information between parties which have not previously interacted, is decentralized, censorship resistance, does not rely on DNS, and allows domains to be assembled into multiple trust groupings.

3. Ethereum Registration Authority

3.1 Architecture

The Ethereum Registration Authority (ERA) system of contracts are Solidity contracts [20] which reside on an Ethereum blockchain, typically Ethereum MainNet. Figure 1 shows the ERA system in the context of Enterprise Ethereum Clients. To establish a sidechain:

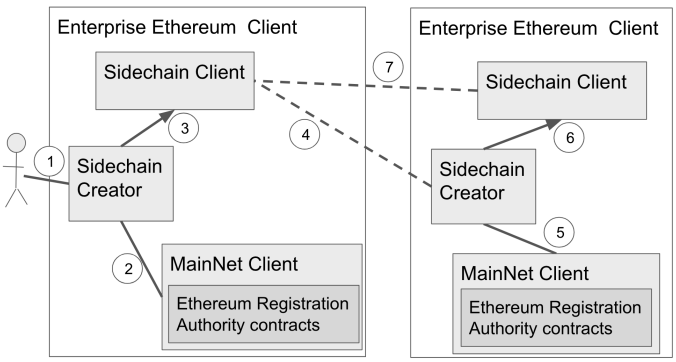

Figure 1: ERA System Context

1. The user calls the Sidechain Creator specifying a list of domain names representing organisations that will participate in the sidechain.
2. The Sidechain Creator reads bootstrap information from the ERA system of contracts, passing in a list of domain names and

receiving bootstrap information for the Enterprise Ethereum Client(s) belonging to each organisation. The bootstrap information could be IP addresses, public encryption keys, public verification keys and Ethereum accounts.

3. The Sidechain Creator instantiates a Sidechain Client, passing in the bootstrap information.
4. The Sidechain Client establishes a connection to the Sidechain Creator for an Enterprise Ethereum Client specified in the bootstrap information, passing domain names of organisations participating in the sidechain. Assuming the permissioning has been set appropriately, the Sidechain Creator accepts the connection.
5. This second Sidechain Creator reads bootstrap information from the ERA system of contracts, passing in a list of domain names and receiving bootstrap information for the Enterprise Ethereum Client(s) belonging to each organisation.
6. The second Sidechain Creator instantiates a Sidechain Client, passing in the bootstrap information.
7. This second Sidechain Client establishes a connection with the first Sidechain Client, thus establishing the Ethereum Private Sidechain between the two Enterprise Ethereum Clients.

As the ERA system of contracts are on Ethereum MainNet, both of the Enterprise Ethereum Clients could be certain of the integrity and authenticity of the bootstrap information stored in the contracts. The Sidechain Creators for each Enterprise Ethereum Clients was able to securely fetch bootstrap information based on the domain names of organisations that were participating in the sidechain.

The ERA system of contracts consists of Domain Information contracts which hold information for one or more domains, ERA contracts which hold look-up information to map between domain names and Delegate ERAs and/or Domain Information contracts, and Finder contracts which allow domain names to be resolved and domain information to be located using a set of Root ERA contracts.

Contracts for Root ERAs and Delegate ERAs are identical, thus allowing a Delegate ERA to operate as a Root ERA in some contexts and as a delegate in other contexts. This could be useful in situations such as when within an organisation, the organisation's own ERA contract should be considered a Root ERA whilst other organisations may wish to locate the organisation's ERA contract via an ERA contract they trust, thus treating the organisation's ERA contract as a delegate.

A Delegate ERA contract can be listed with any number of Root ERA contracts. Doing this means that a Delegate ERA contract can be locatable via multiple paths, thus not tying the Delegate ERA to one Root ERA, and thus reducing the risk of censorship of the Delegate ERA by a single Root ERA.

Domain Information contracts hold domain information for one or more domains. Holding the domain information for multiple domains in one contract reduces the gas cost as fewer contracts need to be deployed. Additionally, information which is relevant to all sub-domains can be stored efficiently in the contract.

Figure 2 shows an example ERA set-up. In the example, there are two root registration authorities, **A** and **B**. Root ERA **A** has an

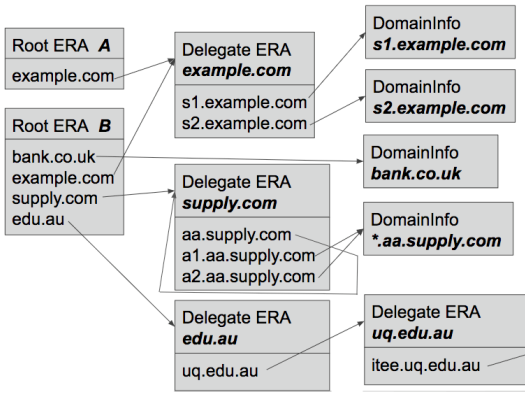

Figure 2: Root and Delegate ERAs and Domain Information

entry for example.com. Example.com is also listed in root ERA **B**. As such, applications which need to access information about example.com or any of its sub-domains could use Root ERA **A** or Root ERA **B**. Example.com operates a delegate ERA contract. It has two sub-domains listed s1.example.com and s2.example.com. The entries for each sub-domain point to separate Domain Information contracts which hold the domain specific information. Bank.co.uk is only listed in Root ERA **B** and does not operate a delegate registration authority. Its entry in Root ERA **B** points directly at the Domain Information contract for bank.co.uk. Supply.com operates a delegate ERA. It has a sub-domain aa.supply.com listed. This sub-domain indicates the delegate ERA for its sub-domains is the supply.com ERA. As such, a1.aa.supply.com and a2.aa.supply.com are listed in the supply.com ERA. Sub-domains a1.aa.supply.com and a2.aa.supply.com use the same Domain Information contract to store their domain information.

In Figure 2, edu.au could map to a government department of education that operates a Delegate ERA. Each university, for example University of Queensland would operate their own Delegate ERA, for example uq.edu.au. The university could delegate the configuration of the Domain Information contracts to the departments within the university. As such, the administrators responsible for the domain itee.uq.edu.au would operate their own Domain Information contract.

Figure 3 shows the ERA system in the context of an Ethereum Sidechain Creator. The sidechain creator needs to determine a value from the Domain Information contract for s2.example.com and bank.co.uk. It uses a Finder contract to resolve the Domain Information contract to use for each domain and fetches the value based on the supplied key. Functions in the Finder contract only

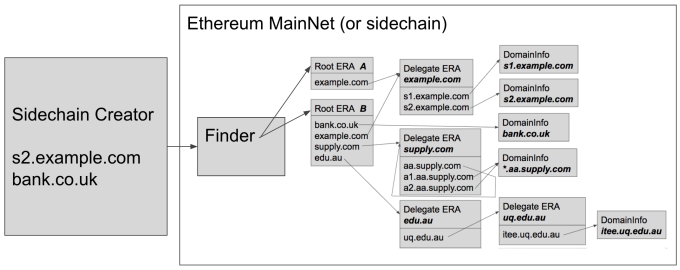

Figure 3: Finder Contract Resolving Domains

fetch information from the local copy of the Ethereum Client's state database, and as such do not incur any gas cost. Additionally, as these functions do not update the state of the ERA or Domain Information contracts, the functions can be complex without risking a vulnerability which could be exploited to alter the state of the ERA or Domain Information. An additional feature of the Finder contract is that it allows hundreds of function calls to the ERA and Domain Information contracts to be wrapped into a single function call, thus reducing the number of calls between application and Ethereum Client, thus reducing the latency between searching for data and locating it.

3.2 Implementation

This section describes some design choices of the ERA reference implementation [5]. The ERA contract holds information indexed using the Keccak-256 message digest of the domain name. The rationale for using the Keccak-256 message digest of the domain, rather than the domain itself is to both minimise the gas usage and make the gas usage deterministic. Ethereum uses a 32-byte word size, the size of the Keccak-256 message digest. When values are put into a map, the key is digested using Keccak-256 inside the Ethereum Virtual Machine. However, if the key value is longer than one word long, multiple digest calls must be made to digest the value, with each call increasing the gas cost. Keccak-256 digesting domain names which have variable length off-chain and submitting the thirty-two-byte digest value ensures the gas cost is fixed as the key is guaranteed to only be one word long.

In the ERA contract, each domain name entry can have an address of a Delegate ERA contract, an address of a Domain Information contract, and the address of the domain owner. This means that for each domain it can have both domain information and sub-domains.

The data type used within the ERA and Domain Information contracts for storing data is a, "mapping". The gas used for storing and retrieving data from a mapping is not dependant on the number of entries in the mapping. As such, restricting data storage in the ERA contracts to this data type means that the gas usage is the same independent of the number of entries.

The Domain Information contracts hold information in a key – value mapping. The value is an array of bytes, the format of which depends on the key. The key is the Keccak-256 message digest of the raw key. The raw key has the format: Domain ":" Key Type. The domain can be a domain name (a1.aa.supply.com) or a wildcard (*), meaning the value for the key type applies to all domains in the Domain Information contract if domain specific values are not specified.

Standard Key Types have been specified in the documentation of the ERA reference implementation [5]. User defined Key Types can be specified as a reverse domain name appended with dot separated names. For example, pegasys.tech has defined a Key Type tech.pegasys.sc.enc which is the public encryption key to be used for an Enterprise Ethereum node.

3.3 Gas Usage Evaluation

Gas is the fee charged for each instruction executed in Ethereum. Different instructions are charged different amounts of gas, with the fees reflecting the economic cost of executing the instruction. Users specify the price they are prepared to pay for the gas in each of their transactions. Miners preferentially include transactions in blocks which are configured to pay the highest gas price. As such, transactions which are submitted with a higher gas price are more likely to be included in a given block.

The Ethereum Gas Station [21] publishes live statistics on how quickly transactions will be processed based on the gas price specified for a transaction. For example, on August 28, 2018, it showed that some miners would process transactions at 0.6 gwei, however only three blocks in the past 200 had included transactions at this price. If a user was prepared to pay 2.6 gwei, then their transaction was likely to be processed within the next two blocks. Given an average block time of 15 seconds, this translates to transactions possibly being processed sometime in the next fifty minutes for gas prices of 0.6 gwei or with a high degree of probability processed in the next thirty seconds for gas prices of 2.6 gwei. This range of confirmation time versus gas price is shown in Figure 4. Confirmation times for gas prices below 1 gwei are either very high, or the transaction does not get mined at all. As the gas price increases, the confirmation time decreases. At 3 gwei, the transaction is likely to go into the next block. As such, there is no benefit to users for offering to pay gas prices above 3 gwei as the transaction is already likely to be mined as soon as it can be. In

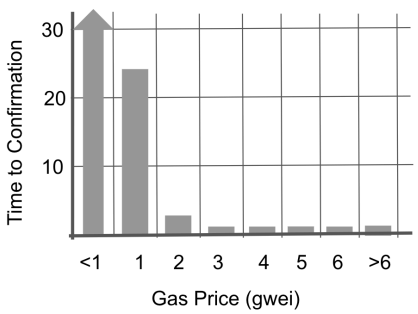

Figure 4: Transaction Confirmation Time versus Gas Price,
Source data: Ethereum Gas Station [21]

times of high network load, significantly higher gas prices may need to be paid to ensure transactions are mined in a timely manner.

The gas usage for the reference implementation of the Ethereum Registration Authority contract is shown in Table 1 and the gas usage for the reference implementation of the Domain Information contract is shown in Table 2. The values reflect the fact that the base cost of all transactions in Ethereum is 21,000 gas, that each write to a new storage location costs 20,000 gas, that subsequent writes to storage locations costs 5,000 gas, that gas is refunded for deleting storage locations, and that most other common functions cost relatively little compared to the cost of data storage. In the tables, the gas cost to US$ conversion is calculated based on an Ether price of US$284, and the fact that 1 Ether is 10^9 gwei.

Table 1. Gas cost for calls to ERA_v1.sol implementation.

Function Call	Scenario	Gas Used	US$[1]
ERA_v1	Contract deployment.	961758	0.164
addUpdateDomain	Set: delegate ERA address, domain information address, and domain owner address. First write.	87663	0.064
	Set: delegate ERA address, domain information address, and domain owner address. Subsequent writes.	42419	0.031
	Set: delegate ERA address only. First Write.	45693	0.033
	Set: delegate ERA address only. Subsequent writes.	30693	0.022
	Set: domain information address only. First Write.	45757	0.034
	Set: domain information address only. Subsequent writes.	30757	0.023
	Set: domain owner address only. First Write.	46909	0.035
	Set: domain owner address only. Subsequent writes.	31665	0.023
removeDomain	Domain to be removed exists.	20099	0.014

Table 2. Gas cost for calls to DomainInfo_v1.sol implementation

Function Call	Scenario	Gas Used	US$[2]
DomainInfo_v1	Contract deployment.	532678	0.091
setValue	Write a value which is between 1 and 31 bytes long (one word). First write.	45240	0.033
	Write a value which is between 1 and 31 bytes long (one word). Subsequent writes.	35297	0.026
	Write a value which is 32 bytes long (two words). First write.	67455	0.050
	Write a value which is 32 bytes long (two words). Subsequent writes.	37455	0.027
	Write a value which is between 33 and 64 bytes long (three words). First write.	87653	0.065
	Write a value which is between 33 and 64 bytes long (three words). Subsequent writes.	42653	0.031
	Write a value which is between 65 and 96 bytes long (four words). First write.	109899	0.081
	Write a value which is between 65 and 96 bytes long (four words). Subsequent writes.	49899	0.037

Assumes Ether price of US$284, 0.6 gwei for contract deployment and 2.6 gwei for other transactions.

For transactions which are not time critical and involve a lot of gas, such as contract deployment, it makes sense to specify a low gas price such as 0.6 gwei. At this price, the Ethereum Registration Authority contract could be deployed for US$0.164 and the Domain Information contract could be deployed for US$0.091. For transactions which are time critical such as changing a value in a Domain Information contract which represents the public key to be used, a higher gas cost should be paid. Setting an ECC 256-bit public key which uses 96 bytes, with a gas price of 2.6 gwei, on the Domain Information contract would cost US$0.081 if the storage locations occupied by the key had not previously been used and US$0.037 if they had.

4. Discussion

The goal of the ERA system is to allow bootstrap information to be determined securely to allow Ethereum Private Sidechains to be established on demand between parties that have not previously interacted. This section analyses whether the ERA system delivers on this goal and analyses its security.

4.1 Establishing Trust on Demand for Free

A user of the ERA system who wishes to establish a sidechain could determine the domain names of organisations which are going to participant in the sidechain and use the ERA system to resolve the Domain Information contract to be used for each domain. They could fetch the bootstrap information they need from the Domain Information contracts. As the user would be viewing existing information, and not submitting transactions, they would interact with the blockchain state on their local Ethereum node. As such, fetching this information would not cost anything.

4.2 Root ERA Business Model

The ERA system is likely to work only if ERAs are incentivized to behave well. There are likely to be two types of Root ERAs: Commercial and Regulated. A Regulated Root ERA would be an ERA which listed organisations that were members of a consortium, an association or other grouping. This type of ERA could be a central bank that listed all of the accredited banks within its jurisdiction, a professional body which listed member organisations, or a government regulator that listed accredited organisations. This type of Root ERA is incentivized to operate their ERA correctly to ensure they are held in high regard both by their member organisations and by users of the ERA system.

Commercial Root ERAs will charge organisations money to list with them. This type of ERA is similar to commercial root X.509 certificate authorities. They are incentivized to operate their ERA correctly to ensure organisations will wish to continue to be listed by them and to ensure users will continue to trust them.

4.3 Trusting Ethereum

The authenticity and integrity of the information stored in the Ethereum network is assured by the algorithms used by Ethereum and the diversity of miners. Users sign the transactions they submit

to the Ethereum network. As such, miners and users can verify the authenticity of transactions. The three largest Proof of Work consensus algorithm miners would need to collude to mount a 51% attack [22] which would allow for the history of the blockchain to be re-written. As such, the information in the blockchain cannot be modified. Given this combination of authenticity and integrity, the Ethereum MainNet is deemed to have strong non-repudiation properties.

Though the Ethereum system typically cannot be modified, after a re-entrancy bug was exploited in the DAO attack [23], the system was modified to reverse the results of the attack. Doing this caused some to question trust in blockchain systems and Ethereum in particular [24]. However, this type of history rewrite appears unlikely to occur again. Despite a bug in the Parity Wallet contract which resulted in hundreds of millions of dollars of funds becoming inaccessible, proposals to alter history to restore the funds have been refused [25] [26].

4.4 Trusting Domain Information

ERAs are responsible for ensuring that the domain names they list are owned by the entities purporting to own them. They are responsible for ensuring the Ethereum account which is used as the domain owner is owned by the entity. Doing this links the domain name, the Ethereum account, and the real world entity. Typically, ERAs will need to undertake a Know Your Customer (KYC) audit and potentially an Anti-Money Laundering (AML) audit prior listing the domain.

An ERA which did not exercise due diligence prior to listing domains would be sanctioned. Users would not trust the ERA to provide information. Domain owners would not want to list on such an ERA. If the ERA was a delegate ERA, then Root ERAs may refuse to list the delegate ERA.

4.5 Malicious ERA and Existing Domains

The ERA Solidity contracts have been written such that once a domain has been registered, the owner of the domain is the only entity which can update the domain's listing, to change the Delegate ERA contract address, the Domain Information contract address, or the address of the owner of the domain. However, the ERA owner could delete a domain entry, create a new malicious domain entry with the same name, and then at some later point change the domain entry back to the original, thus temporarily providing malicious information to users. An ERA acting in this way could be detected by users observing changes to the ERA contract state and Finder results. Additionally, the actions of the malicious ERA would be recorded in the blockchain. An ERA acting in this way would be sanctioned.

4.6 Malicious Registration of Domains

A malicious ERA could maliciously list domain names and set the Delegate ERA contract address and Domain Information contract address to values of their choice. Similar to the abuse-case described above, this nefarious conduct could easily be detected, could not be later concealed, and would lead to the ERA being sanctioned.

4.7 Centralization

A concern with any authority system is that it leads to centralization. With the ERA system, owners of domains can choose to list their domains directly with any number of Root ERAs. They could operate a Delegate ERA and list that with any number of Root ERAs. An owner of an ERA could choose to not list it with any Root ERA. In this case, users would need to trust the ERA directly. Users of ERAs can choose to trust any number of Root ERAs. Users could choose to trust only specific Delegate ERAs. This flexibility of trust, where users can choose which Root ERAs to trust and domain owners can choose which Root ERAs to list in, means this solution can be decentralized.

In some consortium networks, for instance in an inter-banking scenario, a central bank may insist on operating the one and only Root ERA that the banks will use for the purposes of Ethereum Private Sidechains between banks. In this scenario, the consortium chooses to operate with a single Root ERA, recognising that it is a centralisation point.

Another aspect of decentralization of the ERA system is that domain owners can update their own information. They can choose what types of information to store in their Domain Information contracts. This is in contrast to the existing DNS system in which many Domain Registrars exercise control over what information can be stored in domain entries [13].

4.8 Censorship

If a Root ERA delisted a domain or a Delegate ERA, then the domain information for the domain would no longer be accessible by users which only trusted that ERA. Users could directly access the domain's Delegate ERA or Domain Information contracts. Additionally, the domain could be listed with multiple Root ERAs, thus allowing users to find the domain information via an alternative path. Given this, the ERA system has multiple ways of addressing censorship.

5. Conclusion

The Ethereum Registration Authority system allows Ethereum Private Sidechains to be established on demand between Enterprise Ethereum Clients operated by organisations that have not previously interacted. The system overcomes the limitations of previous techniques, providing organisations with a secure, decentralized, censorship resistant mechanism for storing information on Ethereum MainNet such that the information can be located using domain names and can be grouped according to different trust levels and different trust relationships. In particular the system allows users to obtain bootstrap information such as IP addresses, cryptographic verification keys and Ethereum addresses based on organisations' domain names.

A reference implementation of the Ethereum Registration Authority system of contracts has been developed which uses the same amount of gas, independent of the length of domain name and number of entries in the system. The implementation has been shown to be conservative with gas usage. The system can resolve

domains and retrieve information based on a single function call, thus reducing latency and application complexity.

More work should be undertaken to explore Root ERA incentivization models and to standardize the Domain Information contract Key Types to promote interoperability between applications and registrars using the ERA system. This is particularly important as it is possible that the ERA system may be used for storing a wide variety of data, beyond just Ethereum Private Sidechain bootstrap information.

References

[1] P. Robinson, "Requirements for Ethereum Private Sidechains", Arxiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1806.09834. [Accessed: 07- Jul- 2018].

[2] Enterprise Ethereum Alliance, "Enterprise Ethereum Client Specification 1.0". [Online]. Available: https://entethalliance.org/resources/. [Accessed: 01-Jun-2018].

[3] G. Wood, "Ethereum: A Secure Decentralized Generalised Transaction Ledger", 2016. [Online]. Available: https://github.com/ethereum/yellowpaper. [Accessed: 25-Nov-2016].

[4] X. Xu et al., "A Taxonomy of Blockchain-Based Systems for Architecture Design," presented at the 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, 2017. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7930224&isnumber=7930182. [Accessed: 1-Aug-2018].

[5] P. Robinson, "Ethereum Registration Authority Github Repository", 2018. [Online]. Available: https://github.com/ConsenSys/EthereumRegistrationAuthority. [Accessed: 01-Sep-2018].

[6] P. Nielsen, "Quorum Github Repository", 2018. [Online]. Available: https://github.com/jpmorganchase/quorum. [Accessed: 04-Mar-2018].

[7] Ethereum Foundation, "Go Ethereum Github Repository", 2018. [Online]. Available: https://github.com/ethereum/go-ethereum [Accessed: 4-Mar-2018].

[8] Hyperledger Fabric 1.1 Documentation. 2018. [Online]. Available: http://hyperledger-fabric.readthedocs.io/en/release-1.1/getting_started.html. [Accessed: 5-May-2018].

[9] Corda Documentation. 2018. [Online]. Available: https://docs.corda.net/. [Accessed: 10-May 2018].

[10] V. Buterin, "The Meaning of Decentralization", Medium, 2018. [Online]. Available: https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274. [Accessed: 03- Jan- 2018].

[11] D. Clark, "The design philosophy of the DARPA internet protocols", SIGCOMM Comput. Commun. Rev., vol. 18, no. 4, pp. 106-114, 1988.

[12] Protocol Labs, "IPFS Docs", IPFS, 2018. [Online]. Available: https://ipfs.io/docs/. [Accessed: 01- Sep- 2018].

[13] "Configure OpenAlias to more easily share your Monero address", Monero.how, 2018. [Online]. Available: https://www.monero.how/tutorial-configure-openalias-to-receive-monero. [Accessed: 01- Sep- 2018].

[14] T. Chung, "Why DNSSEC deployment remains so low | APNIC Blog", APNIC Blog, 2017. [Online]. Available: https://blog.apnic.net/2017/12/06/dnssec-deployment-remains-low/.

[15] "DNSSEC Measurement Maps", Stats.labs.apnic.net, 2018. [Online]. Available: https://stats.labs.apnic.net/dnssec. [Accessed: 07- Jul- 2018].

[16] "Namecoin", Namecoin.org, 2018. [Online]. Available: https://namecoin.org/. [Accessed: 03- Apr- 2018].

[17] J. Redman, "Onename Drops Namecoin, Switches to Bitcoin", Cointelegraph, 2015. [Online]. Available: https://cointelegraph.com/news/onename-drops-namecoin-switches-to-bitcoin.

[18] J. McDonald, "EthDNS: an Ethereum backend for the Domain Name System", Medium, 2018. [Online]. Available: https://medium.com/@jgm.orinoco/ethdns-an-ethereum-backend-for-the-domain-name-system-d52dabd904b3.

[19] "Ethereum Name Service", Ens.domains, 2018. [Online]. Available: https://ens.domains/. [Accessed: 12- Jan- 2018].

[20] Ethereum Foundation, "Solidity", 2018. [Online]. Available: http://solidity.readthedocs.io/en/latest/ [Accessed: 10-Oct-2018].

[21] "ETH Gas Station", Ethgasstation.info, 2018. [Online]. Available: https://ethgasstation.info/. [Accessed: 03- Aug- 2018].

[22] "Top Miners over the last 24h", etherchain.org, 2018. [Online]. Available: https://www.etherchain.org/charts/topMiners

[23] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)", Berlin, Heidelberg, 2017, pp. 164-186: Springer Berlin Heidelberg.

[24] E. Spode, "The great cryptocurrency heist", Aeon.co, 2017. [Onbline]. Available: https://aeon.co/essays/trust-the-inside-story-of-the-rise-and-fall-of-ethereum . [Accessed: 13-Oct-2018].

[25] N. Johnson, "On Parity's proposed changes to SELFDESTRUCT behaviour", Medium.com, 2018. Available: https://medium.com/@weka/on-paritys-proposed-changes-to-selfdestruct-behaviour-c3f0e5bc0f49 . [Accessed: 13-Oct-2018].

[26] A. Schoedon, "EIP 999", 2018. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-999.md . [Accessed: 13-Oct-2018].

.