Journal of
Theoretical and Applied Electronic Commerce Research

MDPI

Article

Blockchain-Based Address Alias System

Norbert Bodziony, Paweł Jemioło *, Krzysztof Kluza and Marek R. Ogiela

AGH University of Science and Technology, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, al. A. Mickiewicza 30, 30-059 Krakow, Poland; norbert@student.agh.edu.pl (N.B.); kluza@agh.edu.pl (K.K.); mogiela@agh.edu.pl (M.R.O.)
* Correspondence: pawljmlo@agh.edu.pl

Abstract: In recent years, blockchains systems have seen massive adoption in retail and enterprise environments. Cryptocurrencies become more widely adopted, and many online businesses have decided to add the most popular ones, like Bitcoin or Ethereum, next to Visa or Mastercard payments. Due to the decentralized nature of blockchain-based systems, there is no possible way to revert confirmed transactions. It may result in losses caused by human error or poor design of the user interface. We created a cryptocurrency wallet with a full on-chain solution for aliasing accounts and tokens to improve user experience and avoid unnecessary errors. The aliasing system consists of a number of smart contracts deployed on top of the blockchain network that give the ability to register aliases to accounts and tokens and use them instead of opaque addresses. Our solution shows how performant modern blockchains are and presents a way of building fully decentralized applications that can compete with centralized ones in terms of performance.

Keywords: blockchain; cryptocurrency; wallet; aliasing system; Solana

Citation: Bodziony, N.; Jemioło, P.; Kluza, K.; Ogiela, M.R. Blockchain-Based Address Alias System. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 1280–1296. https://doi.org/10.3390/jtaer16050072

Academic Editor: Jani Merikivi

Received: 17 February 2021
Accepted: 11 April 2021
Published: 13 April 2021

1. Introduction

Blockchain [1,2] space is one of the most rapidly growing ecosystems right now. There are various patterns concerning blockchain-based applications [3,4], and many companies, like Coinbase or Binance, quickly reached the status of *unicorns* [5]. The popularity of these technologies has made many companies add *blockchain* or *bitcoin* to their names [6]. Thanks to such attributes as ease and affordability of transactions, tamperproof, and public ledger, blockchain-based cryptocurrencies provide trust through technology [7]. Moreover, cryptocurrencies gain more traction in the retail space and significant corporations often treat Bitcoin [8] as an alternative to gold and other inflation-resistant assets [9].

Digital currencies are considered as the next step in monetary evolution [10]. The Office of the Comptroller of the Currency of the USA has recently announced that federally chartered banks may use stablecoins for transactions [11]. At the same time, many other countries' central banks are preparing to launch their version of cryptocurrencies.

Stablecoins [12,13] are an excellent use case of blockchain technology, allowing for almost instant transfers with minimal fees. They guarantee that the transaction's value will not change, as is the case in other cryptocurrencies. In contrast, traditional bank transfers often take days to arrive, while international transfers lose values on the way due to currency exchanges.

The superior technology of blockchain over traditional methods is often not enough to outweigh potential losses caused by the finality of mistakes [14]. One of the common mistakes is sending a transaction to a wrong address or directly to a smart contract, which results in an irreversible loss of funds. Blockchain addresses are often opaque and long, making them nearly impossible to remember or insert without a copy and paste feature. Even small mistakes may end up in sending transactions to the wrong address [15]. Most of the new users of cryptocurrencies [16,17] often meet with steep learning curves and have to watch out for potential mistakes that they may not even seem valid, like the one mentioned above [18].

This article aims to showcase possible solutions to strictly defining aliases for addresses in human-readable form. The important part is that solutions have to be on chains instead of user-defined aliases that are usually stored only in the user cache. That way, new wallets, exchanges, or other application can quickly query blockchain smart contract to get all registered aliases. Moreover, users will be able to use the same aliases across the entire ecosystem without doing any extra work.

Popularized alias solutions suffer from high fees [19], long confirmation times, and a lack of support for auto-complete features. Our solution aims to solve all of the above issues while keeping the system decentralized and permissioned.

The presented proof of concept was deployed on the Solana blockchain. It is an entirely new blockchain that is highly cost- and performance-optimized to allow near-instant transactions [20]. In functionality, Solana resembles Ethereum [21], and most of the applications working on Ethereum can be easily migrated to Solana [22]. We created a comparison of main key parameters and features of both blockchain systems.

The rest of the paper is structured as follows: Section 2 introduces the necessary terms and presents the currently available solution to address aliasing deployed on the blockchains. While Section 3 focuses on our solution to account aliasing systems, Section 4 shows how tokens work on the Solana network. In Section 5, we present a solution for aliasing tokens. Section 6 presents performed tests of the implemented solution. Next, Section 7 presents data and results, while, in Section 8, we explain potential vulnerabilities and problems. Section 9 provides conclusions and future work.

2. The State of the Art

Blockchain is a convenient and novel way of storing information in a decentralized way that ensures consistency and validity across all network participants. Blockchain technologies enable new innovative solutions in various areas [23–25]. It is the core technology behind most successful cryptocurrencies, e.g., Bitcoin [26,27]. In recent years, blockchain technology has been rapidly evolving and introduces more and more functionalities to its users. Blockchain systems can be divided based on multiple criteria, but the most prevalent one is the consensus mechanism. The field is currently dominated by two solutions [28]: Proof of Work (Bitcoin, Etherum) and Proof of Stake (Solana).

Current blockchain-based systems, like Ethereum or Solana, keep the core principles of Bitcoin but drastically expand their capabilities by introducing smart contracts [29–31]. Smart contracts [30] enable specifying business logic implemented on top of blockchain infrastructure. Completely automated and once deployed on the network, smart contracts do not require maintenance. It is a handy tool for creating communication between multiple parties that do not trust each other. Users can communicate with the blockchain and perform actions using transactions. Each transaction is signed by a user private key to ensure its validity and is broadcasted to the entire network of nodes for validation.

Transactions [32] can be simple operations, like moving money from one address to another, or more complicated ones that involve smart contracts, like borrowing money. These operations are usually performed with cryptocurrency wallets [33,34] that contain users private key used for signing transactions. There are several common solutions (Cryptocurrency wallets: https://trustwallet.com/ accessed on: 1 February 2021, https://metamask.io/ accessed on: 1 February 2021).

Many current cryptocurrency wallets or exchanges (Cryptocurrency wallet: https://electrum.org/ accessed on: 1 February 2021, Cryptocurrency exchange: https://binance.com/ accessed on: 1 February 2021) allow for aliasing specific accounts and using these aliases instead of addresses when creating transactions. This kind of aliases can only be used by users who created them and cannot be accessed using multiple exchanges or wallets.

Ethereum Name Service (ENS) (Ethereum Name Service main page: https://ens.domains/ accessed on: 1 February 2021) is a public naming system built on top of the Ethereum blockchain. It aims to solve the addresses' complexity and length by mapping

them into human-readable aliases, e.g., `norbert.eth`. ENS, in its functionality, is similar to the Internet's Domain Name Service (DNS) [35]. Names registered in ENS have top-level domains but can also register additional subdomains.

Creating a name to address mapping seems like a necessary step for most blockchains to reach wide adoption. Besides solving the accessibility of blockchain addresses for casual users, companies and developers may also benefit from them.

A smart contract, once deployed, is immu [29–31]. One can mitigate this by mapping contract address to name and updating address once the new version is available. This migration will be smooth and will not require even the involvement of users. Some smart contracts that users should not use can also be marked to prevent potential loss of funds. Name service could act as a tool for bringing revenue to sustain the blockchain system. Registration of names could require an inconsiderable fee that can be then distributed.

The current design of DNS is centralized and monopolized by Internet Corporation for Assigned Names and Number (ICANN) [36] which controls the creation of new Top Level Domains and assignment of IP addresses. Additionally, all Internet users need to establish a connection with Internet Service Providers (ISP) before accessing the Internet. This allows ISPs to have surveillance over traffic and possibly restrict or censor some resources [37].

Namecoin [38] is a cryptocurrency based on Bitcoin as also the first attempt at creating a blockchain-based DNS alternative. In its design, Namecoin provides a system of resolving name to value rather than implementing domain-based hierarchical structure. Names registered using Namecoin contain virtual.bit [39] top-level domain. The system provided the same functionalities as traditional DNS, like creating, renewing, or transferring, but in a completely decentralized as premissioned way.

There are designs of systems that enable temporary aliases to be used by various applications in a cryptocurrency transaction [40]. In that solution, the temporary aliases might be assigned from a generated pool of aliases, which can be valid for a specific number of transactions or time.

Agostinho et al. [41] proposed Wallet Domain Name System (WDNS) architecture which handles blockchain wallets and contracts enabling users to manage their domains.

Although the system of domains or aliases might protect users from executing erroneous transactions, it might make the procedure even more susceptible to privacy issues [42], e.g., profiling techniques [43] to deanonymize cryptocurrencies [44].

There are several issues with the mentioned technologies. Namecoin [38] suffers from outdated Bitcoin technology that is not fit for purpose and the necessity of running a specialized node to connect with the network. After all, Namecoin is just a modified Bitcoin code. At the same time, Bitcoin is still actively developed and includes many improvements that Namecoin is missing. ENS [35] and our solution do not suffer from this problem. These systems are built on already established networks, Ethereum and Solana, respectively. Development and improvements of mentioned networks will also positively impact applications built on top of them. Ethereum network is currently overloaded by the amount of traffic that is using the network [22]. It causes that the transaction fee for registering the ENS domain is sometimes many times larger than the actual registration fee. The main point of blockchain fees is to prevent transaction spam. Ethereum's fees have risen drastically over recent months due to increase in popularity of blockchain technology and no longer act as spam filter but tool for arbitrage and front running since Ethereum network is constantly bottle necked. Low fees are essential for creating systems that we can interact frequently and expect almost instant feedback. On the contrary, Solana is prepared for more traffic than Ethereum is experiencing, making it easier to interact with programs, like ENS.

3. Presented Approach

This section describes how account addresses are used and what methods of error prevention they implement. We explain how smart contracts operate on Solana networks

and point out some key differences with the Ethereum blockchain. Lastly, we discuss how we leveraged Solana's smart contract to create a fully on-chain aliasing system.

Solana is much more user-friendly than other networks (see Table 1), both in terms of transaction fees and confirmation speed. The presented solution could be adapted to any blockchain platform that supports smart contracts, but Solana is best fit for this kind of use case thanks to high performance, low fees and ability to change ownership of account. Ethereum transactions have variable costs impacted by network congestion and complexity of invoked smart contracts that can reach tens of dollars for simple transfer or even hundreds for exchanging tokens (see: https://ycharts.com/indicators/ethereum_average_transaction_fee accessed on: 4 February 2021). Solana offers a simple static near-zero fee for all transactions. Moreover, we can bundle multiple transactions together and send them as one reducing the fee even more.

Concerning the decentralized app infrastructure, the presented solution is placed in smart contract and user interface layers, which is visualized in Figure 1.

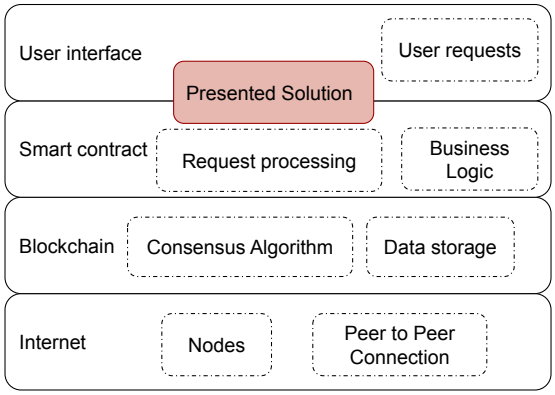

Figure 1. Decentralized app infrastructure.

The interaction with the alias system for users is straightforward. Let us call our example user Alice. Alice's first step is to get a test token that will be used to pay for the network and alias system fee. This can be done on our application by clicking the button Airdrop. Alice can then register her Name by clicking the button Register Alias and filling the alias field, and sending the transaction. If the alias is already taken, the error message will be shown. A User that wants to send tokens to Alice will now have the ability to use her alias instead of a blockchain address.

Users will be able to directly interact with the alias system using created User Interface to register an alias for their address, register alias for token or just send a transaction using aliases instead of addresses of specific accounts. Detailed flow of interaction is presented later in the paper and visualized in Figures 2 and 3.

3.1. Blockchain Addresses

Addresses on the blockchain are used to unambiguously specificity the target of the operation [45]. Both regular account and smart contract have a unique address that one can use to interact with them. It is worth mentioning that one does not create addresses when creating cryptocurrency wallet or accounts. Addresses are already there, and it is a private key to one of them that is generated [46]. Due to the space of possible addresses (2^{160} in the case of Bitcoin), it is close to impossible to generate the same key twice.

As stated before, one of the common mistakes done by new users of blockchain technology is making a typo in the recipient address or sending a transaction to the completely wrong address. Once sent to the wrong address, their money is lost forever [47]. To avoid it, developers introduced checksummed addresses [48,49] that prevent typos.

For example, in the case of Ethereum, the standard address is presented below:

`0xedfca068ed063a856f20bb629e7d03de3149f92b,`

whereas checksummed looks differently:

`0xEdfcA068ED063a856f20BB629e7D03De3149f92B`.

Notice that the first address includes only lowercase letters. Checksummed address contains both uppercase and lowercase letters. Changing any of the address characters invalidates checksum, and in case of sending a transaction to this address, it is wrong. Unfortunately, one cannot force users to use only checksummed addresses, so mistakes due to typos still happen [47].

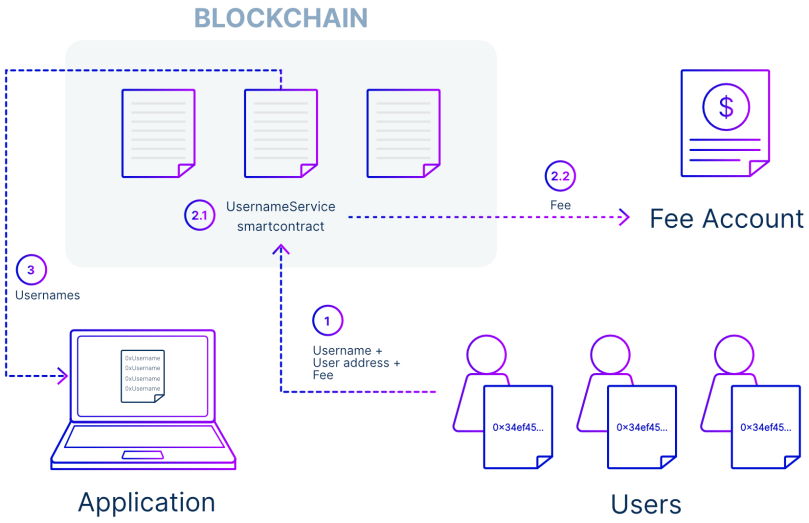

Figure 2. Workflow of Account Aliasing System.

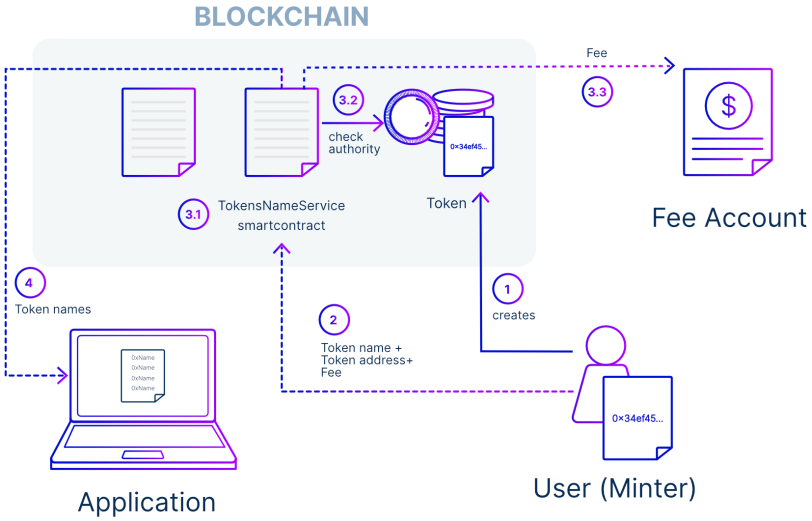

Figure 3. Workflow of Tokens Name Service.

Solana's primary addresses work the same way as described above, but there is a difference in addresses used by smart contracts [22]. Solana's smart contracts do not contain state but can own data accounts. Data accounts have predefined space that smart contract can use to modify. This approach adds some complexity to way addresses, other than native coins, works. Instead of just using one address to move coins, as is the case in Ethereum [50], one needs to create *sub-accounts* for each coin. It is similar to having separate accounts in the banks for different currencies.

Fortunately, this approach comes with mistake prevention. Because moving coins require that *sub-accounts* exist, it is close to impossible to send it to the wrong one by mistake. If one makes a typo in the address, this transaction will not be valid, yet it still does not solve the problem of long opaque addresses.

3.2. Smart Contracts on Solana

Solana's smart contracts have much unique design when compared to other platforms (Blockchains supporting smart contracts: https://ethereum.org/ accessed on: 1 February 2021), https://eos.io/ accessed on: 1 February 2021) supporting them. For reasons of clarity, we focus on the difference between Solana's and Ethereum's smart contracts.

The first significant difference is that smart contracts on Solana do not hold any state. Once deployed, the code is immutable, and the state of the contract can be stored in accounts owned by this contract. These accounts are often called data accounts. Data accounts have a predefined amount of memory up to 10 MB that can be used for storing, and account creators need to pay for allocating memory. In comparison, on Ethereum, the data is stored in smart contract itself, making development easier since one does not have to allocate memory or worry about limits [50]. However, it causes considerable reductions in performance. Solana claims to be orders of magnitude faster than Ethereum [20].

Secondly, smart contracts have a single point of entry in which one passes all necessary parameters for smart contract invocation [22]. One can specify a list of accounts that might be accessed from smart contract and additional instruction data in parameters. One can not access any accounts that are not provided in parameters. Ethereum allows direct invocations of methods specified in the smart contract and can access all accounts or smart contracts without additional steps [50].

Lastly, Solana's smart contracts support parallel runtime execution, which is a feature that is unique for only this blockchain, allowing for concurrency of transactions that do not have overlapping input accounts passed in parameters [22].

All mentioned differences are compiled in Table 1.

Table 1. Comparison of Solana and Ethereum.

	Solana	**Ethereum**
Transaction Throughput	50,000 tps	15 tps
Transaction Fee	0.00001 $	10 $
Transaction Finality	0.6s (1 block)	8 min (30 blocks)
Consensus Algorithm	Proof of Stake	Proof of Work
Scalability with hardware	Yes	No
Supports smart contract	Yes	Yes
Difficulty of development	High	Low
Support for concurrency	Yes	No

Green = pros, red = cons, orange = neutral.

3.3. Other Blockchains

The blockchain ecosystem is rich in networks that support smart contracts. Unfortunately, most of them are just modified versions of Ethereum, e.g., Tron, Binance Smart chain, or Ethereum Classic, and did not gain enough traction compared to Ethereum. Often, other networks are vulnerable to multiple attack vectors, like double-spent (Ethereum Classic double-spent attack: https://coingeek.com/over-1m-double-spent-in-latest-ethereum-classic-51-attack/ accessed on: 2 February 2021).

For our solution, Solana was chosen, instead of other Ethereum-like networks, as it offers an entirely different approach, architecture, and possibilities.

3.4. Account Aliasing System

Our motivation lies in providing casual users an interface to enable features, like autocomplete or notifications when a new user registers a name. Storing names in a human-readable format includes some difficulties in comparison to storing them as hashed values. Hashed values have a constant length. On the contrary, using aliases forces setting a specific range of length that all aliases must match.

In Solana's case, data storage is quite different from other blockchains [22]. Each byte of memory that one uses needs to be paid, so registering aliases on Solana can benefit from using a non-constant length of memory to store aliases.

The system needs to detect if somebody wants to register the same name again; this operation can be quickly done on Ethereum blockchain by using `mapping` method:

```
mapping(address => bytes) public alias.
```

Having all this in mind, we propose a system that enables managing user account. First, each alias is stored in separate data accounts that are dynamically created when users send a transaction that triggers the register function on the blockchain. If one wants to validate if a specific alias is taken, one needs to access all existing data accounts created, which is currently impossible.

A simple solution to this problem is not validating aliases when registering them and providing information about ordering them, and moving validation on aliases off-chain instead. It means each user validates all addresses themselves, improving privacy and security. It requires using an additional data account that acts as a counter and increments with each newly registered user. The counter's value is appended to the recently registered user, so one with a lower counter is selected for conflicting aliases. Instead of passing all existing accounts to smart contract during invocation, the account counter may be utilized when registering new users.

We also proposed an additional feature—a fee charged registering a new user to prevent the system from spam abuse. Although Solana is not as convenient as Ethereum, and a value cannot be added when invoking smart contract transaction, we resolved it by moving the first funds to data account, which stores registered users counter. During the invocation, smart contract first checks if the account contains funds and then move them to specific address as payment for registration.

Each of these small operations, e.g., creating a data account, moving funds, is a separate atomic transaction. However, they can be bundled together into one notable transaction with sub-transactions to ensure correct order and that no one will interfere.

Figure 2 presents how smart contracts are used for interactions. Firstly (step 1), users send transactions with registration data that includes `Username`, `User address` and `Fee`. the transaction then is processed by smart contract (step 2.1) and the transaction fee is sent to `Fee Account` (step 2.2) after a successful registration. Third-party applications can query smart contracts (step 3) to receive all registered usernames and listen for new events representing new registrations. Smart contracts do not require any maintenance and are accessible by all participants in a fair and decentralized manner.

4. Tokens on Solana

Solana, besides its native token SOL (Solana's blockchain native token information: https://solana.com/tokens), does not include any token by default. Each new token created on Solana is a new smart contract. Currently, the most widespread implementation of tokens is the SPL (Solana Program Library) token program [51], and it is widely accepted as a standard implementation in the ecosystem. The token program supports a wide range of features, like freezing, minting, and transferring.

Each token can also have a particular type of authorities:

- Mint authority—responsible for the creation of new tokens.
- Freeze authority—responsible for freezing and thawing balances of accounts.

Tokens do not support naming them during creation, so the only way to connect the address of a specific token to its name is via some off-chain solution (Solana's transactions explorer: https://explorer.solana.com/).

5. Tokens Name Service

This service has a similar design to the Account Aliasing System but works in quite a different way. the simplified version is presented in Figure 3. Token names are not unique, so there is no need to keep comparison or ordering.

To register the token, the user needs to be `Minter` (an account that got minter authority). This field is initialized during the creation of the token (step 1). Registration (step 2), besides token name and address, requires a small fee passed to the fee account after a successful registration (step 3.3). During token registration (step 3.1), smart contract checks if the transaction's sender is precisely the token `Minter` by checking the token's mint authority (step 3.2). The system can pull all registered tokens directly from a smart contract (step 4) and offer human-readable aliasing of tokens for its users.

6. Tests

The performance and speed of blockchain are essential if one wants to deliver similar centralized systems. To validate the proposed name service usability, we tested confirmation speed for registering usernames and token names. Tests were conducted for two types of confirmations on Solana network called commitment. Commitments that we used are listed below.

- `Max`—the transaction is included in a block that is recognized as finalized.
- `SingleGossip`—used for creating transactions in series; recommended in the documentation [52]. This type of commitment is much faster than type `Max`.

All tests were implemented using typescript programming language based on functionalities implemented in Nebula Wallet project (see our GitHub page: https://github.com/Nebula-Wallet accessed on: 2 February 2021) to make them as close to real-world use as possible. Test system specification:

- Operating System: `Ubuntu 20.10`.
- Processor: `Ryzen 7 4800hs`.
- RAM: `16 GB 3200 MHz`.
- Connection: `100 MBps-download, 5 MBps-upload, LAN` (Tested with certified provider. See: https://www.speedtest.pl/wynik/278389151 accessed on: 5 February 2021).

The second test that we performed was pulling token names and account aliases directly from the blockchain. Some external source can index this type of data, so in a real-world scenario, this data could be provided from a different source than a blockchain node. However, for this article's purpose, we will be querying this data directly from Solana blockchains nodes via RPC.

All data about aliases queried from the blockchain were structured as an array of bytes, so users' side will need additional time to process all entries. This time was measured during experiments. The test was performed on the `testnet network` of Solana blockchain. It is worth mentioning that it is possible to subscribe for specific events that occur on blockchain and get updates about name service in real-time without downloading its entire state.

6.1. Account Alias Registration

Registering username will be performed in one bundled transaction that includes three sub-transactions:

- Creation of data account.
- Transferring fee to data account.
- Registration of username.

We measured the time that was needed for the transaction to reach a specific commitment. Tests were conducted in 10 samples for each commitment.

6.2. Token Alias Registration

Similar to the previous test, we used multiple sub-transactions connected into one. This transaction included:

- Creation of token.
- Creation of data account.
- Transfer fee to data account.
- Registration of token name.

Again, we measured the time needed for sending transactions to `singleGossip` and `max` commitment.

6.3. Fetching Alias Data from Blockchain

We performed fetching registered users and tokens for three types of size groups 10, 100, and 1000 instances. Each test included ten samples, and we measured elapsed time since sending requests to receiving responses from blockchain nodes.

6.4. Survey

To validate our solution's usability, we created a survey that gathers valuable data and user feedback. The survey was based on System Usability Scale (SUS) [53] with an additional question about familiarity with cryptocurrency. Instruction (see Table A1) and items (see Table A2) can be found in the Appendix of this paper. The survey was conducted via Internet, and, to gather responses from the participants, we used Google Forms. Each question used the Likert Scale [54] to quantify user response.

7. Results

7.1. Account and Token Alias Registration

Figure 4 shows that time needed to confirm transactions registering tokens and accounts can rival with centralized systems in case of using `singleGossip` commitment.

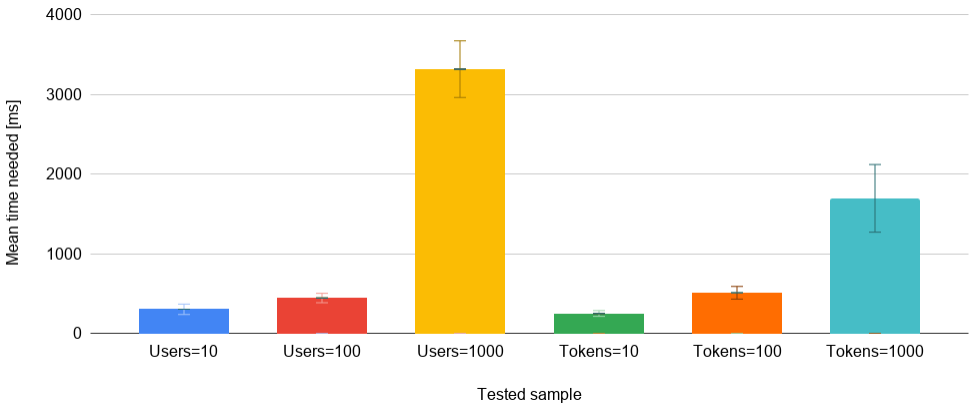

Figure 4. User and token registration transfer times.

Getting a confirmation of a transaction in just 2–4 s seems to be quite an accomplishment for blockchain technology. If we compare it with the most popular blockchains, even getting one confirmation may take minutes in the case of Bitcoin or about 15 s on Ethereum [55]. Commitment `Max` requires much more time than `singleGossip`. However, it ensures that specific transaction is confirmed permanently, which in case of Proof of Work blockchain type is hard to estimate because they are vulnerable to reorganizations.

Most common thresholds introduced by cryptocurrency exchanges [55,56] are:

- 3 confirmations for Bitcoin—about 30 min.
- 30 confirmations for Ethereum and ERC20 tokens—about 8 min.

7.2. Fetching Alias Data from Blockchain

Figure 5 presents times of querying users' aliases directly from nodes. For each size group, there were ten tests. We can see that time needed for query data increases close to linearly with the rising number of entities.

Similarly to users' aliases, Figure 5 presents times measured during tests of querying aliases of tokens. Querying tokens seems to take a little longer than querying users. the correlation between the number of entries and time still is close to linear.

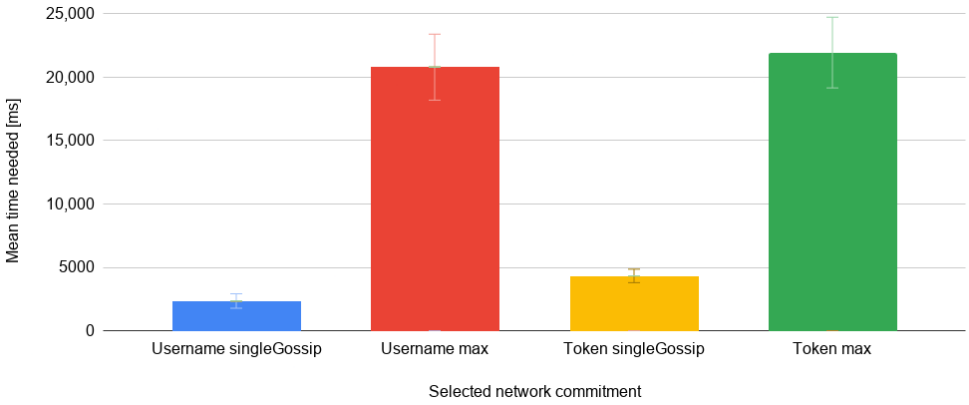

Figure 5. Registered users and tokens query data.

Exponential growth in entries on the blockchain does not result in the exponential time of retrieving. However, it still increases significantly, making it probably close to unusable if extrapolated to millions of users and tokens. Fortunately, scaling is much gentler since we can create additional nodes with indexing to provide this data much faster. It is also worth mentioning that autocompleting millions of entries is a task not feasible for web pages. In a real-life example, the web page sends requests for autocomplete suggestions to nodes with unique alias indexing after typing a predefined number of characters.

Key factors important for users interacting with our alias system are speed and cost of interactions. Solana transaction fee is stable for all transactions. Users will be able to perform two interactions pulling existing aliases or registering a new one. Registration time is stable since growing the number of already registered addresses does not increase complexity. Retrieving registered aliases is strictly connected with the number of aliases, network bandwidth and source node performance. Pulling millions of aliases is not a task feasible for user application, but that could be solved by a third party system providing only a limited subset of all aliases (e.g., for auto-completion). The issue of pulling all addresses in some cases could be resolved by using the deterministically generated addresses (see Section 8.2), which enables us to check if an address exists instead of searching for it in a table of all aliases.

7.3. Survey

Twenty-three test users participated in the survey. Based on the responses, we can deduct that most participants did not have strong knowledge about cryptocurrency since the average result in this item equals 2.29 with a standard deviation of 1.04.

We analyzed data about System Usability to grade out our solution based on the Item Benchmarks for System Usability Scale [57]. The average result of user feedback equals 74.09, with a standard deviation of 11.42. These results give our solution a B (A–F scale, where A states for the best usability) grade according to the mentioned benchmark. The results (each item separately) are presented in Figure 6 (the survey items—questions—can be found in Appendix A in Table A2).

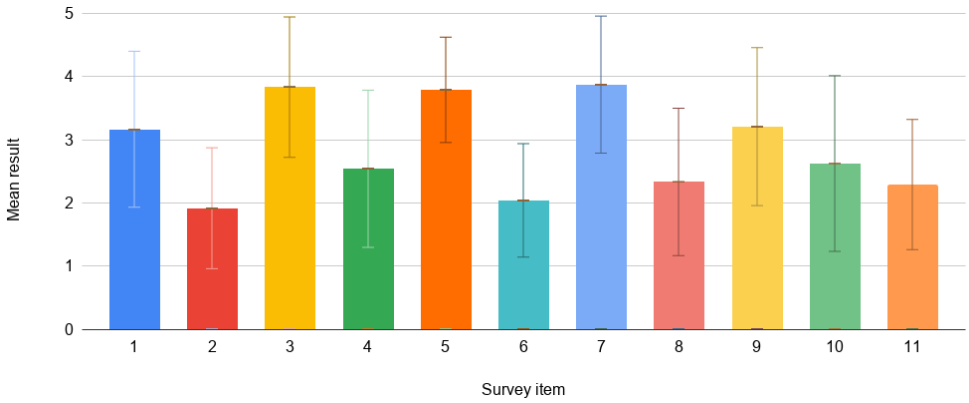

Figure 6. Survey results, each item separately.

8. Discussion

Even if an aliasing system is built and fully accessible in a decentralized way, it is still vulnerable to various attacks. Blockchain transactions are not final immediately [11] after creation and broadcasting. The confirmation time varies depending on the blockchain network we use. The created transaction is still broadcasted to network so it is possible to simulate it or read its data. It opens a way to Frontrun [58] registration message and stealing alias before registration message is confirmed by including malicious message that registers this alias earlier.

Human readable letters have some disadvantages. Many letters can be hard to distinguish and appear the same for casual users. Users can be tricked to send transactions to different aliases using Homograph attack [59]. Exploiters could mimic known aliases of exchanges [60] or smart contracts and point them to different addresses. There is no easy way to handle it. For example, we could specify characters that can be used for registration, but it will cause a limitation of the range of possible names. In the case of similar-looking names, it will always be useful to provide the user with an address to double-check validity. This issue not only affects account aliasing but also token where the exploiter can set the same name of the token as the original one. If aliases systems grow to the point where it is no longer sustainable to pull aliases directly from the blockchain, third-party services that offer indexed sets may manipulate the data provided to users.

Although the presented solution considers aliasing of user addresses, it might also be applied to inter-organizational collaborations in blockchain-enhanced business processes [61–63] and applied within eIDAS-compliant solutions [64].

8.1. Usability

Our solution got a B grade based on the Item Benchmarks for System Usability Scale [57], which is a good result, especially considering that most participants did not know much about cryptocurrency before completing the survey. However, we cannot compare to other solutions as they do not perform usability tests using validated, widely-recognized tools.

8.2. Deterministically Generated Addresses

Blockchain-based accounts are identifiable by public key, but transactions that require signature of this account created are using a private key associated with it. Private key essentially represents ownership of a specific account. Solana blockchain enables to move ownership from private key to specific smart contract, making private key no longer connected with account.

User alias could be used as seed for generating new blockchain account, generated account in this case will always result in the same pair of public and private keys. Ownership of this account is initially under the control of private key but, during alias registration, is moved to smart contract. Smart contract then marks ownership of this account that

now represents an alias to the person that moved its ownership. Once registered in that way, alias is permanently bonded to smart contract since, even if other users generate this account again, private key no longer controls it. This unique architecture allows for static time of access to alias data since we know exactly what specific account to query based on the deterministic address generated using the alias. Verification if the alias is already registered will only involve checking who currently controls the account if its private key, and we are free to register this alias.

This solution is only available on the Solana blockchain and could drastically reduce the time needed to resolve specific aliases. In comparison to ENS and DNS, we no longer need to resolve names from the top domain potentially going through many nodes since all information is accessible from one deterministic address.

8.3. Security

Both DNS and ENS require root nodes to be honest for the system to be secure. In the case of DNS, root nodes usually are controlled by government or public organizations and in the case of ENS, root contract controlled is by multisignature smart contract with keys held by trustworthy individuals from the Ethereum community. Our solution does not define admins or other contracts that control the system. Once deployed, a solution on blockchain lives as a completely permissionless and decentralized system with a static address of smart contract representing the static point of access.

Blockchain-based alternatives are immune to known DNS vulnerabilities, like spoofing or denial of service (DOS) attacks. All data can be validated using a fully synchronized blockchain node.

Table 2 shows the comparison between DNS, ENS, and our solution in aggregated form. As one can see, our solution is similar in many cases to ENS. The critical difference is that ENS requires multiple calls to resolve alias to a specific address. On the contrary, our solution, thanks to deterministically generated addresses, enables one to predict the address of specific alias, making resolving the name a simple operation.

Table 2. Comparison between mentioned name services.

	DNS	ENS	Proposed Solution
Root controlled by 3rd party	Yes	Yes	No
Vulnerable to spoofing	Yes	No	No
Vulnerable to denial of service attack	Yes	No	No
Resolving name	Requires multiple call	Requires multiple calls	One call
Cost of maintenance	High	Low	Low

Green = pros, red = cons, orange = neutral.

Each of these systems is built on top of different architecture, and the majority of differences came from the limitation of the underlying architecture.

9. Conclusions

We have created an utterly on-chain aliasing system for both accounts and tokens that enable highly convenient for the everyday user experience of interacting with the blockchain network. Aliasing systems are based on Solana networks using their version of smart contracts to ensure decentralization and fairness of the system. Users can register address or token in exchange for a small fee that is transferred to a `Fee Account` that can be then used for future development. This solution could have significant implications in terms of users interacting with the blockchain network and applications. With the high adaptation of aliases, it could become a standard way of interaction with opaque blockchain addresses.

Performed tests aimed to show the validity of the use case in a real-world application. We tested times needed for performing registration with different types of confirmations and checked times needed for retrieving data directly from blockchain nodes. Tests show how quickly transactions can be confirmed on the Solana network. Transactions are confirmed in seconds, whereas, in the case of Bitcoin or Ethereum, one is forced to wait minutes. Transactions performed on SingleGossip commitment level can rival even with centralized systems in terms of performance. Directly querying the data shows close to the linear correlation between the number of records and the time needed for retrieving them. However, it can be easily scaled using third party indexing systems.

Additionally, to support our statements, we created a survey to gather feedback about the solution. Most of the participants did not have previously experienced with cryptocurrency, and the survey still presented promising results. However, further tests with the target group and the following statistical analyses are needed.

Modern implementations of blockchain systems, like Solana, present a tremendous opportunity for new types of decentralized applications that do not have to suffer from latency or extensive transaction costs. We have created an alias system for both account and token addresses that any network user can freely register. Account aliases are unique, so, once registered, the alias will always be mapped to one specific address. Registration itself is fast in comparison to other blockchains and can challenge fully centralized applications. Third-party applications can freely query blockchain nodes for information about aliases and use them to enable many convenient user features.

Presented systems can be expanded into multiple directions that ensure profitability, convenience or even act as a *Know your customer (KYC)* system. Aliases could be registered for a specific time frame, similarly as in internet domains. It will ensure a constant inflow of capital for further system development. In addition, aliases will no longer be lost if somebody decides not to use them. Once registration expires, new users will be able to claim aliases.

Currently, all tokens on Solana require a separate account to store them. Registering aliases for each new account is not a perfect solution. We could expand the proposed system to add a layer that provides a new address (mapped to specific tokens connected with already registered aliases). By doing so, when creating a new account for a token, users will be able to connect this address to their alias. Thus, it will highly simplify sending tokens between users. Some aliases could point to more information than just an address. Users could provide additional information, like address, name, and other data. This data could be used for KYC purposes or be validated by some off-chain entity and after confirmation act as synthetic on-chain identity. Such a created on-chain identity record could be used by exchanges, shops, or applications, removing the overhead of passing KYC on each of these platforms.

In the meantime, OAuth authentication becomes the preferred way to login or register. Having one account to use all sorts of applications is simple and convenient. Unfortunately, OAuth providers, like Google, often use users data for internal profits or could censor specific websites. Our solution with some improvements could be used as a decentralized OAuth provider controlled by no one and accessible to everybody.

Due to the decentralized nature of blockchain networks, it is challenging to simulate existing networks' work reliably. In this paper, we have used a test network that aims to be a close copy of an entire network. However, due to multiple factors, like the number of nodes, geolocation, and parameters of nodes, results may vary from the actual network. Thus, additional tests are still required.

Author Contributions: Conceptualization, N.B.; Methodology, N.B.; Software development, N.B.; Writing—original draft, N.B.; Supervision of the project, P.J.; Writing—review & editing, N.B., P.J., K.K.; Funding acquisition, N.B.; Validation, K.K. and M.R.O.; Supervision, M.R.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by AGH UST, and open access fee was covered by Solana Foundation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code for the implementation of the research presented in this paper is openly available online in the code repository: https://github.com/Nebula-Wallet accessed on 13 April 2021.

Acknowledgments: The authors would like to thank the anonymous reviewers for providing valuable comments and suggestions to improve the paper.

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

Appendix A

Table A1 presents the instruction for user evaluation of our proof of concept cryptocurrency wallet solution called Nebula wallet. In Table A2, we presented the Nebula wallet usability survey which was filled in after following the user evaluation instruction. It was adopted from Systema Usability Scale [53].

Table A1. User evaluation instruction.

Nebula wallet is a cryptocurrency wallet that allows you to send and receive cryptocurrency tokens just as you use your bank account to send and receive national currency. Our wallet supports a name service that allows you to register and use simple aliases (names) instead of long account numbers (in our case long addresses). During this survey, you will be testing basic functionalities of the wallet and later fill a short form to express your opinion.
1. Open `https://nebulawallet.com/` accessed on 13 April 2021.
2. Click 'Airdrop' to get test tokens. Your field 'Balance' should increase. Balance represents the amount of cryptocurrency that you currently hold similarly to your bank account balance.
3. Register your own alias by clicking 'Register Alias' button a fill required fields. After submission, 'Register Alias' button should be replaced with the name of your Alias.
4. Send transfer to already registered user by clicking button 'Send'. Your balance should reduce by the amount sent. You've just sent money!
5. Switch to 'Manage Tab' and create new token with the alias. A new token should appear on a token list. We can compare tokens to other types of currencies, like JPY or EUR.
6. Switch to tab Wallet, click 'Add Account' button and select your token. Your created account for token should be visible under 'Tokens' header. We cannot mix multiple currencies, like EUR or USD, in a single account that's why we have a separate address for each of these accounts, but our main account still can manage them.
7. Please fill in the questionnaire.
8. Thank You!

Table A2. Nebula wallet usability questionnaire.

1. I think that I would like to use this website frequently.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
2. I found this website unnecessarily complex.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
3. I thought this website was easy to use.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
4. I think that I would need assistance to be able to use this website.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
5. I found the various functions in this website were well integrated.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
6. I thought there was too much inconsistency in this website.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
7. I would imagine that most people would learn to use this website very quickly.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
8. I found this website very cumbersome/awkward to use.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
9. I felt very confident using this website.						
Strongly Disagree	1	2	3	4	5	Strongly Agree
10. I needed to learn a lot of things before I could get going with this website						
Strongly Disagree	1	2	3	4	5	Strongly Agree
11. I know how to use cryptocurrency.						
Strongly Disagree	1	2	3	4	5	Strongly Agree

References

1. Lu, Y. The blockchain: State-of-the-art and research challenges. *J. Ind. Inf. Integr.* **2019**, *15*, 80–90. [CrossRef]
2. Kaur, A.; Nayyar, A.; Singh, P. Blockchain: A path to the future. *Cryptocurrencies Blockchain Technol. Appl.* **2020**, *2*, 25–42.
3. Xu, X.; Pautasso, C.; Zhu, L.; Lu, Q.; Weber, I. A pattern collection for blockchain-based applications. In Proceedings of the 23rd European Conference on Pattern Languages of Programs, Irsee, Germany, 4–8 July 2018; pp. 1–20.
4. Zielińska, A.; Skowron, M.; Bień, A. The concept of the blockchain technology model use to settle the charging process of an electric vehicle. In Proceedings of the 2019 Applications of Electromagnetics in Modern Engineering and Medicine (PTZE), Janów Podlaski, Poland, 9–12 June 2019; pp. 271–274.
5. Hackett, R. Coinbase Becomes First Bitcoin 'Unicorn'. Available online: https://www.forbes.com/sites/laurashin/2017/08/10/coinbase-becomes-first-crypto-unicorn-raises-100-million-in-funding-amid-ico-craze/ (accessed on 3 February 2021).
6. Jain, A.; Jain, C. Blockchain hysteria: Adding "blockchain" to company's name. *Econ. Lett.* **2019**, *181*, 178–181. [CrossRef]
7. Marella, V.; Upreti, B.; Merikivi, J.; Tuunainen, V.K. Understanding the creation of trust in cryptocurrencies: The case of Bitcoin. *Electron. Mark.* **2020**, 1–13. [CrossRef]
8. Nakamoto, S. Bitcoin: A Peer-To-Peer Electronic Cash System. 2009. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 3 February 2021).
9. Harvey, C.R.; Ramachandran, A.; Santoro, J. DeFi and the Future of Finance. *SSRN 3711777* **2020**. [CrossRef]
10. Viñuela, C.; Sapena, J.; Wandosell, G. The Future of Money and the Central Bank Digital Currency Dilemma. *Sustainability* **2020**, *12*, 9697. [CrossRef]
11. Sharma, R. Federally Chartered Banks Can Use Stablecoins: OCC. Available online: https://www.investopedia.com/federally-chartered-banks-can-use-stablecoins-occ-5094500 (accessed on 3 February 2021).
12. Saito, K.; Iwamura, M. How to make a digital currency on a blockchain stable. *Future Gener. Comput. Syst.* **2019**, *100*, 58–69. [CrossRef]

13. Lyons, R.K.; Viswanath-Natraj, G. *What Keeps Stablecoins Stable?* Technical Report; National Bureau of Economic Research: Cambridge, MA, USA, 2020.
14. Livshits, S.; Novikova, O.; Yudina, N.; Nikolaeva, E.; Katz, D. Possible risks of the development of the digital economy. In Proceedings of the International Conference on Digital Technologies in Logistics and Infrastructure (ICDTLI 2019), St. Petersburg, Russia, 4–5 April 2019; Atlantis Press: Paris, France, 2019; pp. 225–228.
15. Alshamsi, A.; Andras, P. User perception of Bitcoin usability and security across novice users. *Int. J. Hum. Comput. Stud.* **2019**, *126*, 94–110. [CrossRef]
16. Valdeolmillos, D.; Mezquita, Y.; González-Briones, A.; Prieto, J.; Corchado, J.M. Blockchain technology: A review of the current challenges of cryptocurrency. In *International Congress on Blockchain and Applications*; Springer: Cham, Germany, 2019; pp. 153–160.
17. Tama, B.A.; Kweka, B.J.; Park, Y.; Rhee, K.H. A critical review of blockchain and its current applications. In Proceedings of the 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Yogyakarta, Indonesia, 19–21 September 2017; pp. 109–113.
18. Krombholz, K.; Judmayer, A.; Gusenbauer, M.; Weippl, E. The other side of the coin: User experiences with bitcoin security and privacy. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 22–26 February 2016; pp. 555–580.
19. Spain, M.; Foley, S.; Gramoli, V. The Impact of Ethereum Throughput and Fees on Transaction Latency During ICOs. In *International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2019)*; Danos, V., Herlihy, M., Potop-Butucaru, M., Prat, J., Tucci-Piergiovanni, S., Eds.; Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2020; Volume 71, pp. 9:1–9:15. [CrossRef]
20. Yakovenko, A. Tower BFT: Solana's High Performance Implementation of PBFT. Available online: medium.com/solana-labs/tower-bft-solanas-high-performance-implementation-of-pbft-464725911e79 (accessed on 11 November 2020).
21. Vujičić, D.; Jagodić, D.; Ranđić, S. Blockchain technology, bitcoin, and Ethereum: A brief overview. In Proceedings of the 2018 17th International Symposium Infoteh-Jahorina (Infoteh), East Sarajevo, Bosnia and Herzegovina, 21–23 March 2018; pp. 1–6.
22. Solana Sealevel. Available online: https://medium.com/solana-labs/sealevel-parallel-processing-thousands-of-smart-contracts-d814b378192 (accessed on 11 November 2020).
23. Ogiela, M.R.; Ogiela, L. Application of blockchain technologies for secure information management. In *Optics and Photonics for Information Processing XII. International Society for Optics and Photonics*; SPIE: Bellingham, WA, USA, 2018; Volume 10751, p. 107511A.
24. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
25. Wang, X.; Zha, X.; Ni, W.; Liu, R.P.; Guo, Y.J.; Niu, X.; Zheng, K. Survey on blockchain for Internet of Things. *Comput. Commun.* **2019**, *136*, 10–29. [CrossRef]
26. Franco, P. *Understanding Bitcoin: Cryptography Engineering and Economics*; Wiley Online Library: Cornwall, UK, 2015.
27. Rahouti, M.; Xiong, K.; Ghani, N. Bitcoin concepts, threats, and machine-learning security solutions. *IEEE Access* **2018**, *6*, 67189–67205. [CrossRef]
28. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM Sigmetrics Perform. Eval. Rev.* **2014**, *42*, 34–37. [CrossRef]
29. Leka, E.; Selimi, B.; Lamani, L. Systematic literature review of blockchain applications: Smart contracts. In Proceedings of the 2019 International Conference on Information Technologies (InfoTech), Nanjing, China, 9–11 May 2019; pp. 1–3.
30. Ante, L. Smart Contracts on the Blockchain—A Bibliometric Analysis and Review. *Telemat. Informat.* **2021**, *57*, 101519. [CrossRef]
31. Alharby, M.; Van Moorsel, A. Blockchain-based smart contracts: A systematic mapping study. *arXiv* **2017**, arXiv:1710.06372.
32. Beck, R.; Stenum Czepluch, J.; Lollike, N.; Malone, S. Blockchain–the gateway to trust-free cryptographic transactions. In Proceedings of the 2016 European Conference on Information Systems (ECIS), Istanbul, Turkey, 12–15 June 2016.
33. Er-Rajy, L.; El Kiram My, A.; El Ghazouani, M.; Achbarou, O. Blockchain: Bitcoin wallet cryptography security, challenges and countermeasures. *J. Internet Bank. Commer.* **2017**, *22*, 1–29.
34. Dai, W.; Deng, J.; Wang, Q.; Cui, C.; Zou, D.; Jin, H. SBLWT: A secure blockchain lightweight wallet based on trustzone. *IEEE Access* **2018**, *6*, 40638–40648. [CrossRef]
35. Al-Mashhadi, S.; Manickam, S. A brief review of blockchain-based DNS systems. *Int. J. Internet Technol. Secur. Trans.* **2020**, *10*, 420–432. [CrossRef]
36. Wang, X.; Li, K.; Li, H.; Li, Y.; Liang, Z. ConsortiumDNS: A distributed domain name service based on consortium chain. In Proceedings of the 2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Bangkok, Thailand, 18–20 December 2017; pp. 617–620.
37. Zarrin, J.; Wen, P.; Saheer, L.; Zarrin, B. Blockchain for Decentralization of Internet: Prospects, Trends, and Challenges. *arXiv* **2020**, arXiv:2011.01096.
38. Aabid Hussain Ganai, M.A.S. Decentralization of DNS using Blockchain: A Survey. *Int. J. Comput. Sci. Eng.* **2019**, *7*, 2–5.
39. Kalodner, H.A.; Carlsten, M.; Ellenbogen, P.; Bonneau, J.; Narayanan, A. An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design. In Proceedings of the The Workshop on the Economics of Information Security (WEIS), Delft, The Netherlands, 22–23 June 2015; WEIS: Sunbury, PA, USA, 2015.

40. Mokhasi, G.S. Dynamic Cryptocurrency Aliasing, 2020. U.S. Patent 10,614,456, 18 August 2016.
41. Agostinho, B.M.; Pasini, F.B.; Gomes, F.O.; Pinto, A.S.R.; Dantas, M.A.R. An Approach Adopting Ethereum as a Wallet Domain Name System within the Economy of Things Context. In Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), Leicester, UK, 7–10 December 2020; pp. 176–185.
42. Macrinici, D.; Cartofeanu, C.; Gao, S. Smart contract applications within blockchain technology: A systematic mapping study. *Telemat. Informat.* **2018**, *35*, 2337–2354. [CrossRef]
43. Béres, F.; Seres, I.A.; Benczúr, A.A.; Quintyne-Collins, M. Blockchain is Watching You: Profiling and Deanonymizing Ethereum Users. *arXiv* **2020**, arXiv:2005.14051.
44. Gaihre, A.; Pandey, S.; Liu, H. Deanonymizing cryptocurrency with graph learning: The promises and challenges. In Proceedings of the 2019 IEEE Conference on Communications and Network Security (CNS), Washington, DC, USA, 10–12 June 2019; pp. 1–3.
45. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and smart contracts for insurance: Is the technology mature enough? *Future Internet* **2018**, *10*, 20. [CrossRef]
46. Sala, M.; Sogiorno, D.; Taufer, D. A Small Subgroup Attack on Bitcoin Address Generation. *Mathematics* **2020**, *8*, 1645. [CrossRef]
47. Atzei, N.; Bartoletti, M.; Cimoli, T. A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 164–186.
48. Daulay, R.S.A.; Nasution, S.M.; Paryasto, M.W. Realization and addressing analysis in blockchain bitcoin. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Busan, Korea, 25–27 August 2017; Volume 260, p. 012002.
49. Vitalik Buterin, A.V.d.S. Mixed-Case Checksum Address Encoding. Available online: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-55.md (accessed on 12 October 2020).
50. Ethereum Documentation. Available online: https://ethereum.org/en/developers/docs/ (accessed on 1 February 2021).
51. Solana Program Library Documentation. Available online: https://spl.solana.com/ (accessed on 12 October 2020).
52. Solana Documentation. Available online: https://docs.solana.com/ (accessed on 12 October 2020).
53. Brooke, J. Sus: A "quick and dirty'usability. *Usability Eval. Ind.* **1996**, 21, *189*.
54. Likert, R. A technique for the measurement of attitudes. *Arch. Psychol.* **1932**, *22*, 5–55.
55. Kraken Support. Available online: https://support.kraken.com/hc/en-us/articles/ (accessed on 1 February 2021).
56. Coinbase Help. Available online: https://help.coinbase.com/en/coinbase/ (accessed on 12 October 2020).
57. Lewis, J.R.; Sauro, J. The factor structure of the system usability scale. In Proceedings of the International Conference on Human Centered Design, San Diego, CA, USA, 19–24 July 2009; pp. 94–103.
58. Daian, P.; Goldfeder, S.; Kell, T.; Li, Y.; Zhao, X.; Bentov, I.; Breidenbach, L.; Juels, A. Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges. *arXiv* **2019**, arXiv:1904.05234.
59. Holgers, T.; Watson, D.E.; Gribble, S.D. Cutting through the Confusion: A Measurement Study of Homograph Attacks. In *USENIX 2006 Annual Technical Conference Refereed Paper*; USENIX The Advanced Computing Systems Association: Berkeley, CA, USA, 2006.
60. Andryukhin, A.A. Phishing Attacks and Preventions in Blockchain Based Projects. In Proceedings of the 2019 International Conference on Engineering Technologies and Computer Science (EnT), Moscow, Russia, 26–27 March 2019; pp. 15–19. [CrossRef]
61. Müller, M.; Ostern, N.; Rosemann, M. Silver Bullet for All Trust Issues? Blockchain-Based Trust Patterns for Collaborative Business Processes. In Proceedings of the International Conference on Business Process Management, Vienna, Austria, 9–14 September 2020; pp. 3–18.
62. Sturm, C.; Jablonski, S. Interorganizational Process Execution Beyond Ethereum: Road to a Special Purpose Ecosystem. In Proceedings of the Workshops Co-Organized with the 13th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modelling (PoEM 2020), On-Line, Riga, Latvia, 26 November 2020; pp. 68–79.
63. Kampik, T.; Najjar, A. Simulating, off-chain and on-chain: Agent-based simulations in cross-organizational business processes. *Information* **2020**, *11*, 34. [CrossRef]
64. Argento, L.; Buccafurri, F.; Furfaro, A.; Graziano, S.; Guzzo, A.; Lax, G.; Pasqua, F.; Saccà, D. ID-Service: A Blockchain-Based Platform to Support Digital-Identity-Aware Service Accountability. *Appl. Sci.* **2021**, *11*, 165. [CrossRef]