applied sciences

MDPI

Article

Setonix: Blockchain-Based Hierarchy Domain Name System for Web3

Juseong Jeon and Sejin Park *

Department of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; mo66jyp@gmail.com
* Correspondence: baksejin@kmu.ac.kr

Abstract: DNS is an essential component for internet access, but the traditional centralized structure is not suitable for the rapidly growing Web3 ecosystem. Currently, Web3 services rely on Web2-based access paths, revealing limitations such as the Single Point of Failure (SPoF) issue and the compromise of decentralization principles. To address these issues, this paper proposes Setonix, a blockchain-based decentralized DNS. Experimental results of the implemented Setonix showed that it had only a marginal performance difference of up to approximately 5% compared to Google and Cloudflare DNS. Additionally, it demonstrated over 95% lower latency compared to Handshake DNS, proving its high efficiency. Setonix eliminates dependency on centralized access paths and brings the hierarchical address structure of existing domain names into the blockchain network, adhering to legacy systems to minimize user resistance. Additionally, it provides a blockchain-based architecture that supports cross-compatibility between traditional domain names and Web3 domain names through a simple interface. This design significantly enhances the scalability and accessibility of Web3, presenting a new standard for decentralized internet services. By doing so, it helps to establish a foundation that contributes to the growth and innovation of the Web3 ecosystem.

Keywords: blockchain; DNS; Web3; domain name

Citation: Jeon, J.; Park, S. Setonix: Blockchain-Based Hierarchy Domain Name System for Web3. *Appl. Sci.* **2024**, *14*, 11213. https://doi.org/10.3390/app142311213

Academic Editor: Gianluca Lax

Received: 2 November 2024
Revised: 21 November 2024
Accepted: 25 November 2024
Published: 2 December 2024

1. Introduction

Web3 allows users to own the data generated on the web, involving a shift from the traditional server–client model to a serverless topology [1,2]. Therefore, Web3 is generally implemented on a well-known decentralized serverless system, which is based on blockchain networks with a high level of security, to support this serverless topology [3,4]. Currently, Web3 services are offered across various fields, including gaming [5], finance (DeFi) [6,7], non-fungible token (NFT) marketplaces [8], and decentralized cloud storage [9]. These Web3 services, which are being offered in various sectors like Web2, are considered to have high growth potential due to the advantages of serverless topology and their broad applicability across different fields [10].

According to [11], the number of papers published on Web3 has increased more than eightfold in recent years. Additionally, based on the data in [12], it is estimated that by 2030, the Web3 market will grow to approximately USD 16 billion in the U.S. alone and around USD 65 billion globally, indicating significant potential for development. However, despite being recognized for its high growth potential and being built on a serverless blockchain network, Web3 faces the contradictory issue of relying on centralized Web2-based pages as access points for services. The reason for this is that there is currently no platform or program capable of directly supporting access to decentralized web services. Even if one wishes to use traditional web access methods, under the current topology, passing through centralized pathways remains inevitable.

Generally, an essential component in providing users with easy access to web services is the DNS [13]. However, as implied by the term 'server', each server has a centralized

structure with a fixed location. The centralized structure poses a threat to the stability of all Web3.0 services, which are implemented on a blockchain and emphasize decentralization, due to the SPoF issues and potential attack vectors inherent in such a structure [14,15].

Furthermore, in this context, the shift to a serverless topology signifies a departure from the traditional method where servers with fixed physical locations had to be rented or owned to provide web services through the server–client model. As the transition to Web3 progresses, the burden of owning fixed servers decreases, allowing anyone to easily create these services, which ultimately suggests that many Web3 services will emerge [16]. With the explosive growth of Web3 services, the existing DNS structure and the number of servers it relies on are insufficient to support access. As mentioned earlier, the fundamental differences in the structure pose challenges for Web3 access via conventional DNS. Even aside from this issue, the servers responsible for handling the exponentially growing load of Web3 services will not only fail to provide an optimal user experience, but also create significant management challenges for DNS operators.

Here, before formally specifying the problem and proposing a solution, we clarify the terminology to prevent confusion. This helps maintain consistency when referring to blockchain-based DNSs, dedicated domain names, and traditional DNSs like Cloudflare [17] throughout the paper. First, 'conventional domain' refers to domain names like 'google.com' that are commonly in use today. Such domain names are typically served by 'conventional DNS' systems like Cloudflare, but they can also be served by blockchain-based DNSs like Handshake [18], depending on DNS functionality. 'Web3 domain' refers to domain names structured with unconventional TLDs, such as 'superdomain.hdns'. These domains can only be served by blockchain-based DNS and not by conventional DNS. Consequently, registration can only take place on blockchain-based DNSs, where these domains are stored.

1.1. Basic Concepts and Services Relevant to This Paper

In this section, the concepts of conventional DNS and Web3, which are the primary focus of this paper, are first explained to help readers gain a deeper understanding of the content. Additionally, the overall operation and functionality of two existing systems that adopt blockchain-based architectures similar to Setonix are examined prior to the design of Setonix. This provides the foundational knowledge of DNS, Web3, and blockchain-based DNS, facilitating a clearer understanding of the Setonix design section. Furthermore, it aids in interpreting experimental results when these systems are compared with Setonix as blockchain-based DNS solutions.

1.1.1. Operation of Traditional DNSs

The currently operating DNS generally manages each set of letters in a domain, named based on dots ('.'), by dividing them into root, TLD (top-level domain), and SLD (second-level domain) servers. The process of resolving addresses follows a hierarchical structure that recursively goes through root → TLD → SLD [19]. This prevents all requests from being concentrated on a single root server and has the advantage of facilitating the management of subdomains for each domain. Furthermore, this hierarchical system allows for efficient distribution of DNS queries and reduces latency by caching responses at various levels.

1.1.2. Overview of Web3 Systems and the Necessity of Serverless Architecture

Web3 refers to the concept of users owning the data generated on the web, which necessitates a shift in topology from traditional server-based web services to a serverless approach. The reason a serverless model is essential for transferring data ownership from corporations to consumers is that, in the current Web2 model, typical web services are operated by companies. These companies provide services using servers built on significant financial resources. This implies that every user activity—such as writing a blog, browsing, and enjoying content—must pass through these servers. Consequently, server owners can

monitor, record, and reproduce user data, effectively turning user activities into resources for their benefit [20].

1.1.3. Ethereum Name Service (ENS)

As the name suggests, ENS is a DNS operating on the Ethereum platform that simplifies complex Ethereum addresses like 0x123... into a more readable format [21]. Unlike Handshake DNS, ENS is not designed to support a universal address system, so its scope is limited to the Ethereum network. Its actual implementation consists of three main smart contract components: registry, registrar, and resolver.

The registry is the smart contract at the core of ENS resolution, as all ENS resolution requests begin with the registry. It maintains and manages the list of domains, records information such as the resolver, TTL, and owner for each domain, and allows domain owners to update these data upon request. Next, the registrar is divided into the BaseRegistrar and the ETHRegistrarController to reduce the attack surface. The BaseRegistrar is responsible for managing name ownership and transfers, while the Controller handles registration and renewal.

Lastly, the resolver is a component that every name registered in ENS has, and it is responsible for retrieving the information linked to the name, such as actual addresses or avatars. Although users can implement a custom resolver themselves, they typically use a universal resolver known as the public resolver. For a standard forward resolution, a query is first sent to the ENS registry to find the resolver used by the ENS name, and the matching resolver's address is obtained. Once the matching resolver is obtained, another query is sent to the resolver to request the relevant data. For example, when requesting an Ethereum address, the addr() function is used. However, due to its complex structure, design, and multiple stages of resolution, it shows significantly lower performance compared to conventional DNS.

1.1.4. Handshake DNS

Handshake aims to replace the centralized root zone file and root servers, making the root zone censorship-resistant and usable without permission. Additionally, it differs in purpose from ENS, as it seeks to eliminate centralized gatekeepers by addressing criticisms of ICANN [22], which has been viewed as unfair, illegal, anti-competitive, and unrepresentative of diverse internet users [23] since the early days of internet governance. By managing the root zone file, Handshake takes the position of decentralized management of TLDs. Therefore, it includes existing TLDs while also serving its own TLDs. For existing TLDs, Handshake pre-reserves domains for the top 100,000 domains on Alexa, including stakeholders who already own these domains, aiming to replace the existing DNS as much as possible.

The blockchain network that replaces the root zone is based on the Proof of Work (PoW) consensus algorithm and uses the Hashcash function to prevent Sybil attacks and achieve consensus. Domain registration follows an auction process involving bidding, revealing, selecting the highest bidder, and then registering the domain. The payment is determined using the Vickrey auction method, where the winner pays the second-highest bid. By using the Vickrey auction method, participants are encouraged to submit bids that reflect their true value, which helps prevent price sniping and assists in discovering the true market value of domain names. However, due to the adoption of PoW, which has a structurally low TPS, and its complex processes are similar to ENS, its performance is significantly poorer compared to conventional DNS systems.

In domain name resolving, running a full node to resolve names is burdensome for users, so Simplified Payment Verification (SPV) allows lightweight clients to participate in the Handshake network and resolve names. The SPV consists of a p2p layer, authorized name server, recursive server, and non-recursive resolver. The p2p layer synchronizes block headers and verifies name tree proofs, the authorized name server converts on-chain resources into DNS responses, and the recursive server sets its root hints and trust anchors

to its own authoritative root server, rather than ICANNs. Lastly, the non-recursive resolver resides in the authoritative layer and acts as a fallback to ICANN's system. This is used when a reserved top-level domain has not yet been claimed.

2. Challenges with Conventional and Decentralized DNS in Relation to Web3

The previously mentioned issues regarding Web3, Web3 service access methods, DNS, and existing blockchain-based DNS can be summarized as follows:

- **Structural incompatibility between the centralized DNS architecture and the decentralized nature of Web3:** The traditional centralized server–client DNS operates using fixed-location servers in a hierarchical structure. This does not align with the philosophy of Web3, which aims for decentralization and a serverless topology, and thus cannot effectively support Web3 services.
- **Stability threats due to a Single Point of Failure:** Web3 services, which operate on a decentralized basis, rely on centralized Web2 access paths due to the lack of a system that directly supports access to these services, which introduces the Single Point of Failure (SPoF) issue.
- **Scalability limitations of DNS in response to the exponential growth of Web3 services:** Unlike Web2 services, which inherently carry the burden of operating servers at fixed physical locations, Web3 services, with reduced operational burdens, are likely to increase exponentially. As a result, conventional DNS, designed for Web2, will struggle to accommodate this growth, which will not only hinder user experience but also introduce management difficulties from the perspective of DNS operations.
- **Limitations of existing blockchain-based DNS in providing compatibility between Web2 and Web3:** Blockchain-based DNSs designed to provide Web3 services or replace conventional DNSs often fail to offer compatibility between Web2 and Web3. Even when compatibility is provided, poor performance can cause users to face significant resistance in utilizing such services. To address these issues, this study aims to answer the following questions:

    1. How can we provide a platform that supports Web3 service access in a fully decentralized manner, addressing the stability threats posed by current methods?
    2. How can Setonix DNS overcome the scalability issues inherent in conventional DNS?
    3. How can Setonix effectively provide compatibility between Web2 and Web3?

3. Conventional and Decentralized DNS in Relation to Web3

For these reasons, it is crucial that the aforementioned issues be addressed, especially at this point in time when the transition to Web3 is underway. To address this, this paper proposes Setonix, a decentralized DNS based on blockchain, in alignment with Web3. When building a decentralized DNS structure, it offers the following advantages. First, if any node within the network is enabled to perform DNS functions, it enables load distribution, which is more effective compared to the server–client structure. Secondly, since it adopts the same structure as the system that supports Web3, it becomes possible to resolve the SPoF issue, particularly in situations where the Web2 structure serves as an access point and introduces the SPoF problem. Given these advantages, a blockchain-based decentralized structure is a reasonable choice for a DNS aimed at supporting Web3.

First, in this paper, we design the DNS using smart contracts to operate it on a blockchain, which is considered the most representative and widely used decentralized system. When designing, various DNS-related functions such as domain name storage and subdomain assignment are structured hierarchically to facilitate management and resolution. Accessing a DNS designed and uploaded onto a blockchain network in the same address-based manner as traditional systems is challenging. Therefore, an interface system that enables the easy use of Setonix DNS and provides compatibility between Web2 and Web3 on the user side for load distribution is designed and implemented. This system

is integrated with the DNS to optimize the user experience. Through this approach, we can hypothesize how Setonix addresses the previously presented questions:

1. Setonix is designed and implemented as a fully decentralized structure operating on a blockchain to support Web3 service access. This approach aims to resolve the mismatch in the overall system structure caused by the partial use of centralized methods in providing decentralized Web3 services.
2. Setonix DNS is designed to operate on a blockchain that functions through multiple nodes, enabling load distribution across nodes even in rapidly increasing service scenarios. This design offers greater scalability compared to conventional DNS, which operates on centralized servers.
3. By utilizing an interface that interacts with Setonix operating on the user side, the load for providing compatibility can be distributed among users. This approach is expected to deliver higher performance in compatibility compared to existing blockchain-based DNSs.

As outlined in this hypothesis, the purpose of this paper is to demonstrate that Setonix supports fully decentralized access for existing Web3 services while offering significantly lower latency and higher usability compared to existing blockchain-based DNSs. To enhance usability, the primary performance metric naturally emphasized for a DNS is resolving latency, as it is the most perceptible to users. Other DNS performance metrics are excluded from the scope of this study.

Therefore, the designed Setonix DNS conducts resolving latency-focused testing on an Ethereum-based blockchain system. This system supports EVM-based smart contracts written in Solidity, offering a developer-friendly environment for blockchain-based services and system development. Ethereum provides test networks such as Goerli [24] and Sepolia [25] in addition to its main net, making it useful for testing the proposed system. Furthermore, the transition to a PoS consensus protocol in September 2022 and ongoing upgrades through additional 2-layer solutions have continuously improved performance, making it suitable for conducting these tests.

Testing on the main net involves constraints such as gas fees and security risks. Therefore, the system is uploaded to the Sepolia network, one of Ethereum's test networks, which offers a PoS-based environment similar to the main net and is focused on DApp development. The uploaded Setonix DNS is tested for both traditional domain names and Web3 domain names. The results are examined to determine whether it delivers a level of performance suitable for actual user use.

4. Design of Setonix DNS for Supporting Fully Decentralized Web3 Access

In this section, we design a Setonix. First, we establish the considerations necessary for the design and then proceed with the actual design based on these considerations. In the Design Consideration section, we briefly mention the foundational technologies required for the design and the situations where the proposed system is needed, followed by a description of the considerations. Next, in the Architecture section, we present the architecture designed according to the considerations, using diagrams and providing explanations for clarity.

4.1. Design Consideration

The conventional DNS is generally implemented by dividing servers into root, TLD, and SLD, where different parts of each address are stored. Subsequently, queries are called recursively through these servers. Address lookup requests from users are designed to query each server in this recursive manner, sequentially, until the result is returned. The intent behind this implementation is to distribute the load by processing DNS requests across multiple servers rather than a single server. More importantly, it simplifies registration and lookup through an address system separated by dots ('.'). This makes it especially useful for easily verifying ownership in the subdomain part under a domain name.

For example, in the case of Google's domain, the entity that owns the domain name 'google' connected to the TLD '.com' in 'google.com' naturally holds ownership of the associated subdomains under the current address system rules. This allows them to freely create relevant and useful domain names for their services, such as 'map.google.com' or 'drive.google.com'. This means that these subdomains can be created without interference from anyone other than the owner of 'google.com'.

The address system provided by the DNS proposed in this paper must be designed so that users can transition to and use the Web3-specific address system without experiencing significant load. To achieve this, the system follows the principle of implementing an address format similar to the existing one. This ensures ease of use and management through a hierarchical address structure, even on a blockchain-based platform. However, the SLDs, such as .co, .ac, and .go, come in a wide variety, and there is no consistent rule between domains that include SLDs and those that do not. Therefore, to reduce implementation complexity and enhance manageability, this system defines the basic domain format as "domain name + TLD".

Like the conventional DNS, the DNS implemented on a decentralized system must also guarantee ownership of the domain name to the requester based on the registered domain name. However, due to the characteristics of Web3 and decentralization, where user information is not stored in a specific location, this cannot be conducted in the same way as before. Therefore, to enable this functionality, a smart contract that guarantees ownership will be implemented. Users prove their identity through their wallet address, and once ownership is issued via the ownership contract, they can verify their ownership of the domain name.

As mentioned in the introduction, since it is not possible to access a DNS uploaded to the blockchain using traditional methods, an interface is also necessary to provide easy access for users. First, this interface must be implemented as a smart contract capable of interacting with the decentralized DNS uploaded to the blockchain network. Second, to maximize user convenience, it must support all types of domain names without requiring any additional transition process on the user's side. In other words, it should support both conventional domain names used to access Web2 services and separate domain names used to access Web3 services. In summary, a hierarchical naming system that offers great convenience in management and usage is designed to function on a blockchain-based decentralized platform. A separate smart contract is used to assign ownership to each decentralized domain name operating within this system. Additionally, to ensure the smooth use of the DNS that supports this address system, the goal is to integrate a proxy-based interaction tool to deliver the result.

4.2. Architecture

The design is carried out by reflecting the considerations described in Section 4.1. The design follows the sequence in which the intended tasks operate. Therefore, the design is carried out in a top–down manner, starting with specifying the necessary tasks for each component, determining the order of operations for each task, and then defining the elements according to each step. Since the interface, which interacts with the DNS, can only be defined after the DNS is established, the DNS is designed first, followed by the interface.

4.2.1. DNS Part

First, in DNS, ownership management is essential since information about who holds each domain name is needed in every process. Therefore, the design begins with the ownership contract that manages ownership as the top priority, followed by the DNS design incorporating this contract.

The ownership contract that manages ownership is shown in Figure 1 below. The data consist of wallet addresses and domain names, with each domain name being matched in a one-to-many format with those owning wallet addresses. This smart contract operates with a verify ownership function to check if ownership matches for a domain name, and a set

ownership function to match the domain name with a wallet address, storing ownership for the domain name. The fetch ownership function directly returns ownership information for a specific domain name.

Finally, update ownership functions to update the ownership stored as a wallet address for a domain name.

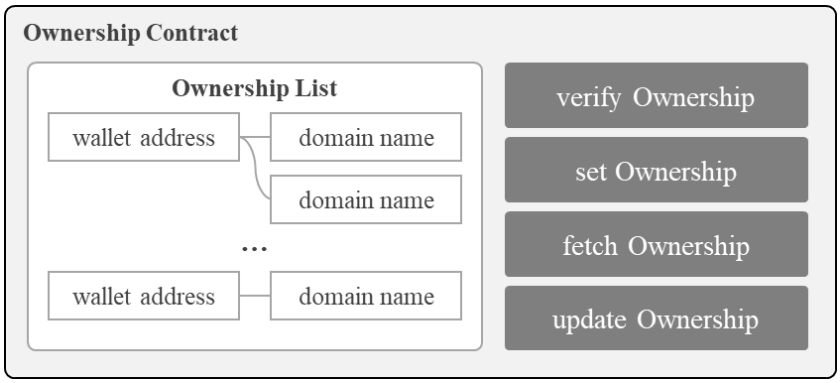

Figure 1. The architecture of the ownership contract, a smart contract that manages ownership of domain names registered in Setonix.

Next, the DNS involves a registration process, which stores a new domain name matched to an address, and a resolver process, which verifies it. Registration must occur first for resolving to be possible, so the design begins with registration, followed by the resolving process. The overall process for registration can be seen in Figure 2 below.

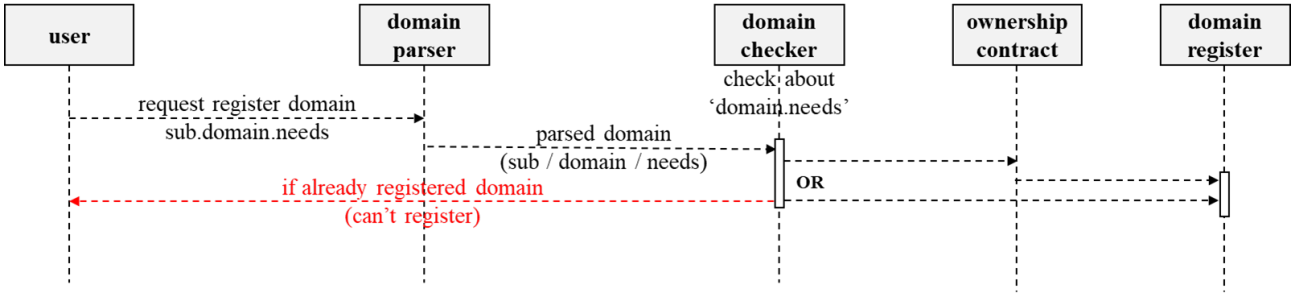

Figure 2. A sequence diagram illustrating the process of registering a domain name in Setonix.

The process follows the arrows as shown in the figure. First, for registration, the registrant's wallet address, the domain name to be registered, and the corresponding address are required. The wallet address is needed to match the owner information with the domain name when executing the ownership registration contract. When the required data are received by the DNS along with the request message, it is first processed by the domain parser and then sent to the domain checker. The domain checker examines the parsed domain to determine whether the domain name is already registered or if it includes a subdomain. If the check confirms that the domain is not registered, it is sent to the ownership contract to assign ownership to the domain first, and then registration proceeds. If only a subdomain is being registered, it is sent directly to the domain register.

In this process, the domain hierarchy structure, designed for convenience in management and use, is included. To explain in detail, take the Web3 domain 'sub.domain.needs' as an example. When this domain is entered for registration, the domain parser splits it by dots ('.'), dividing it into 'sub', 'domain', and 'needs'. The parsed domain is checked by the domain checker in its basic form, 'domain.needs', by grouping the domain name and TLD together, excluding the subdomain 'sub', as defined in the Considerations section.

If the check reveals that the domain is already registered but includes a subdomain (as in the current example), the ownership information issued for that domain is verified. If it

matches, the domain register is executed for the subdomain. However, if no subdomain exists, the same domain name cannot be reissued, so an error message is returned. Finally, if the domain is not registered, a new registration is attempted. Ownership registration is processed for the base domain through the ownership contract. If a subdomain is attached, additional registration is carried out for it as well.

Next is the resolving function, which is requested through an interface described later. This process is much simpler compared to the domain name registration process. The domain resolver within the DNS is designed to return the address matching the requested domain name to the requester. The overall architecture of the Setonix DNS contract, which includes both registration and resolving functions, is shown below in Figure 3.

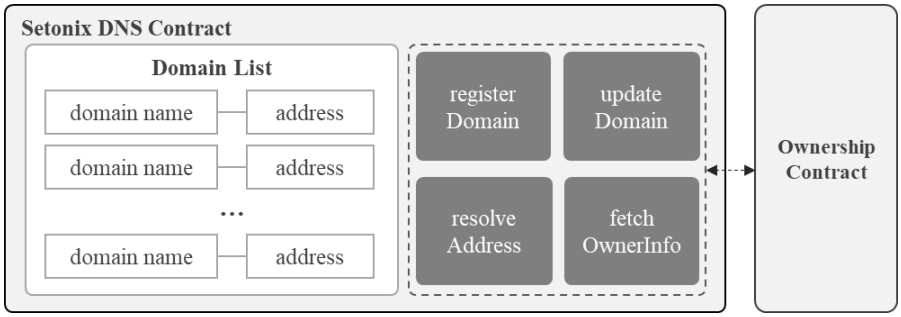

Figure 3. The architecture of the Setonix DNS contract, which is uploaded to the blockchain network and operates as an actual DNS.

As shown in the diagram, functions for updating, directly fetching ownership, and resolving and registering are also included. All these functions operate in close interaction with the ownership contract in Figure 1 due to the need for ownership management.

4.2.2. Interface Part

As mentioned earlier in the Design Consideration section, accessing the DNS uploaded to the blockchain network in the same way as traditional systems presents challenges. To address this, an interface is designed to enable the use of the DNS, and it is packaged together with the DNS as a set to enhance user convenience. Among the various methods available for creating the interface, a proxy was chosen because it allows for installation on the user's device and enables the local resolver to route through the proxy.

The advantage of this approach is that all DNS requests from the user's device can be routed through the proxy, making it adaptable for all services. Additionally, when using a local resolver, the cache it contains can be utilized, which helps to compensate for the potentially higher latency of the blockchain-based DNS compared to conventional DNS.

In line with the considerations mentioned earlier, the system is designed to allow the use of both conventional domain names and dedicated Web3 domain names without requiring a separate DNS switching process for user convenience. To achieve this, a protocol checker is placed within the interface to distinguish between conventional and dedicated domain names (Web3 domains) in the incoming domain resolving requests, applying different resolving methods to each.

Figure 4 illustrates the content described so far, providing a visual representation that makes it easier to understand the compatibility between Web2 and Web3 DNSs. It shows how a request originating from a local resolver passes through the protocol checker within the proxy interface and is categorized into upper and lower pathways. This is made possible by the protocol checker, which is designed to determine the prefix of the received domain name and classify it based on the presence or absence of the prefix. Through this process, it enables seamless switching between Web2 and Web3 domain names without explicit user intervention.

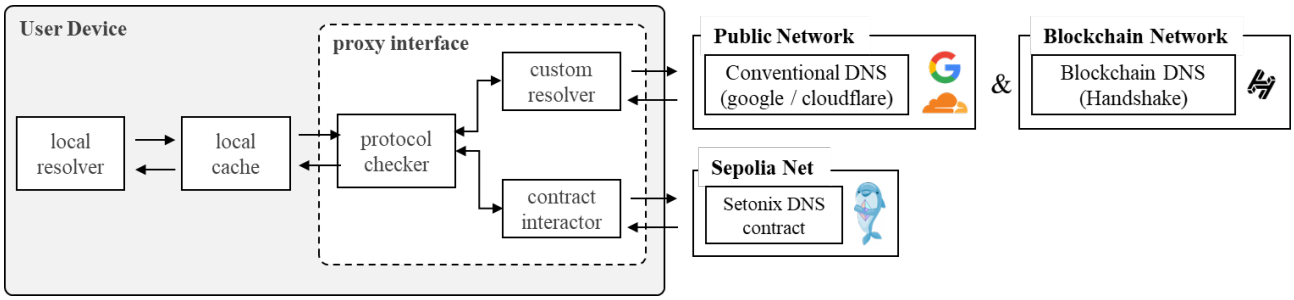

Figure 4. A figure illustrating how the entire Setonix system operates, particularly showing how the protocol checker sends requests to different DNSs through resolvers and interactors. In the case of Handshake, the access point for use is a fixed IP, and it supports processing DNS request packets from standard resolvers. Thus, as depicted in the figure, requests are handled through a general resolver.

The custom resolver is a module that handles conventional domain names categorized by the protocol checker, and it allows the user to choose which existing DNS, such as Google or Cloudflare, to send the request to. The contract interactor is a module used to resolve the remaining custom domain names categorized by the protocol checker. As the name suggests, it is equipped with the functionality to interact with the DNS uploaded to the blockchain network, allowing it to retrieve the address from the domain name–address pairs mapped on the blockchain and to serve the result.

Figure 5 is a sequence diagram that illustrates the entire process, from a domain-name-resolving request to accessing a Web3 service, using all the elements described in the Architecture section. To review the process so far, the user device sends a domain-name-resolving request to the proxy client running on the device for service access. The protocol checker classifies the request and forwards it to the Setonix DNS uploaded on the blockchain for resolution. The Setonix DNS receives the request and searches for the address matching the domain name in its stored list of names and addresses. It then returns the result to the client interface, which sends it back to the user device. The user device, having received the address, uses that information to access the Web3 service.

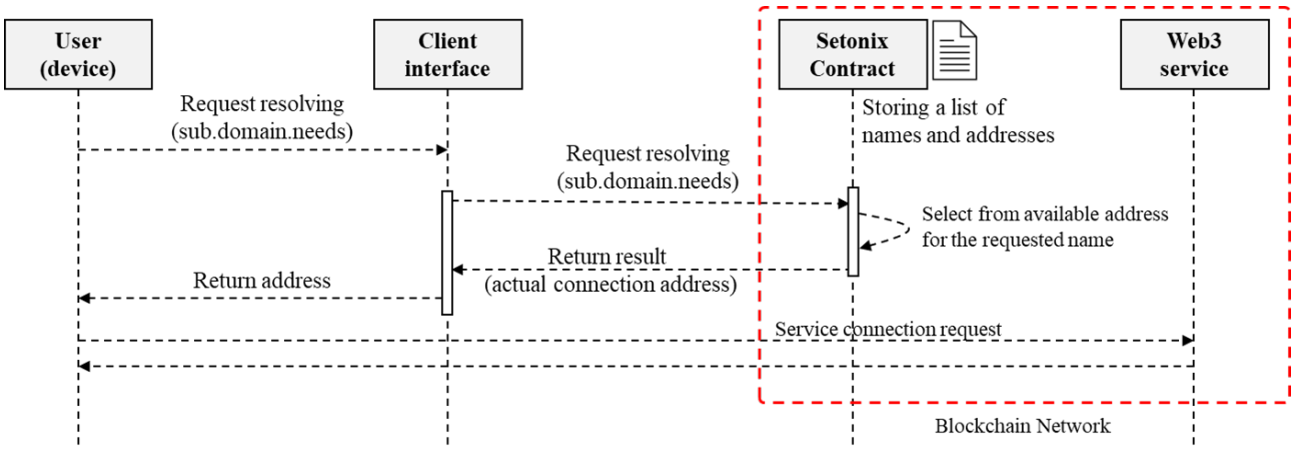

Figure 5. A sequence diagram illustrating the process of a user accessing Web3 services using Setonix DNS.

Access to conventional addresses is handled differently by the client interface. When the user requests an address from the interface, if the protocol checker identifies it as a conventional domain name, the request is sent to the pre-configured conventional DNS. The process then continues by receiving the IP address from the DNS, returning it to the device, and using it to access the service. In other words, the client interface enables a seamless handling of all types of addresses without any DNS-switching processes.

However, performing domain registration through this interface, which focuses on convenience, poses challenges due to the various elements required in the registration process. Therefore, a simple program, either a webpage or a webview-based application, is built to interact with the DNS and handle the registration process.

The design of Setonix to support hierarchical domain names was devised to preserve the efficiency and legacy of traditional Web2 DNSs, minimizing user resistance to new services. At the same time, it maintains convenience while implementing the philosophy of a decentralized Web3. This design overcomes the performance issues and limited user convenience observed in existing blockchain-based DNSs such as ENS and Handshake. In particular, the proxy-based interface and efficient protocol checker enable support for both Web2 and Web3 services, which are challenging to achieve with traditional blockchain-based DNSs. As a result, users can seamlessly access various domain-based services without requiring special transition processes or experiencing resistance. Through this, Setonix provides a solution that significantly enhances Web3 accessibility.

5. Implementation

In this section, the architecture defined in the previous section is first implemented by building the DNS, followed by developing a proxy-based client interface program that users can interact with it, in the same order as described.

5.1. DNS Part

Since the DNS must be uploaded to a blockchain network, as described earlier, the first step is to decide which blockchain to use. In this paper, Ethereum was selected due to its high activity and widespread use, as well as its support for development on the blockchain. Additionally, development was carried out using Solidity, the smart contract language that is supported on this blockchain.

The domain parser, domain register, and domain resolver described in the architecture were each implemented as functions, while the domain checker was implemented as logic operating within the actual register function. Once all the operations shown in Figure 3 are completed and the final registration is performed in the smart contract, the data are stored in a map structure. The reason for using a map structure is that it has an internal structure similar to a hash table, allowing data to be stored efficiently. Moreover, retrieving the stored data can be performed just as quickly as insertion, with the same O(1) time complexity. Since it is implemented based on a map structure, the function for resolving is also designed to retrieve matching data from the map when requested or to return an error message if no match is found.

5.2. Interface Part

The interface is implemented in Go language and uses a package that allows the creation of DNS packets with the desired data to provide responses to requests from the user device. The client interface opens and waits on port 53 of localhost. The user runs the program and configures the local resolver on their device to send DNS requests to localhost. Once this configuration is complete, the interface receives the requests on port 53. The part that interacts with the actual DNS is implemented based on the go-ethereum package, and it is designed to interact with the Ethereum network either through direct RPC communication or via blockchain access API services such as Infura [26].

The previously mentioned domain name registration page was implemented using HTML, CSS, and JavaScript, with the Web3.js library used in JavaScript for interacting with Ethereum. The web functionality itself was implemented using Go's net/http package, allowing users to run the webpage locally, connect it to their Metamask wallet [27], pay a small gas fee, and register the desired domain name.

6. Experimental Results

In this section, the hierarchical structure of Setonix is uploaded to the actual blockchain, and a comparison is made with other DNSs and blockchain-based DNSs through the proxy interface. The blockchain used was the Sepolia network, which is Ethereum's testnet, and the user (or machine) sending the requests through the installed proxy interface used a Windows OS PC for compatibility. The detailed specifications of the PC are a 13th Gen Intel® Core™ i9-13900 CPU, 64 GB RAM, Realtek Gaming 2.5 GbE Family Controller #2 NIC, and a 1 Gbps bandwidth network.

Domain registration on the Sepolia Testnet incurred a cost of approximately USD 0.5 for registering a single domain name, as no additional fees were required by policy. While this cost may vary slightly depending on the type of network, congestion, or the length of the domain name, it is significantly more affordable compared to other systems. For instance, ENS requires an annual fee of USD 640 plus additional network fees for domains of 3-character length, and Handshake DNS transactions typically average around USD 20 but can vary widely based on rarity [28].

The experiment first evaluated three DNSs: Google and Cloudflare as representative traditional DNSs, and Handshake as a blockchain-based DNS capable of resolving traditional domain names. The most noticeable aspect for users when using DNSs is the latency during resolving to access services. Therefore, resolving latency was chosen as a key test parameter and measured accordingly. In Figure 6, the graph on the left shows the results of resolving domain names using the local resolver on a Windows PC, while the graph on the right shows the latency when resolving each DNS through the custom resolver within the proposed proxy-based interface.

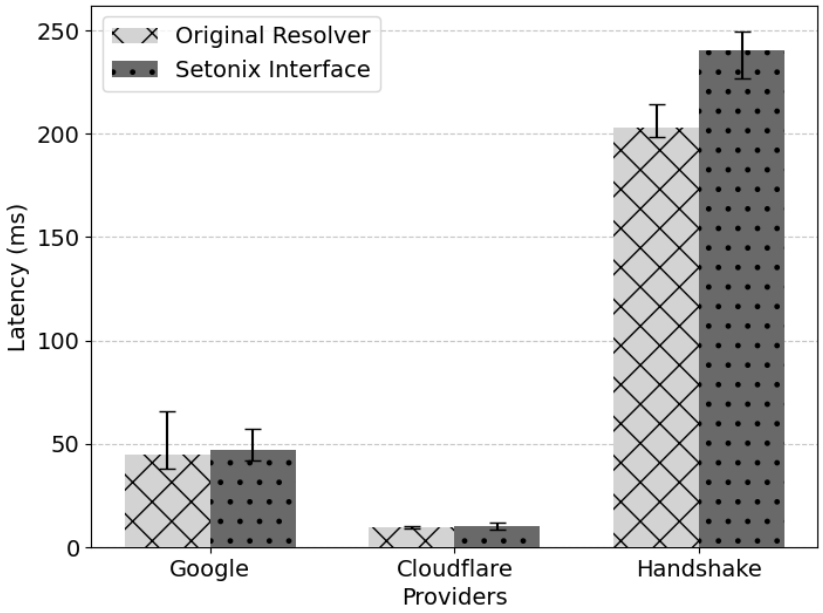

Figure 6. This represents the latency differences when resolving through the Original Resolver and Setonix's proxy-based interface for each DNS. Each error bar indicates the maximum and minimum values for the corresponding data.

The resolving latency measurement experiment was conducted through the following design and process:

- Domain selection: The top 100 resolvable domains were chosen using Cloudflare's top 1 M websites list.
- The resolving process compared two methods for each DNS: the Original Resolver method and the Setonix Interface method, which includes the resolver function. In the Original Resolver method, a local resolver was used to resolve domain names. In con-

trast, the Setonix Interface method employed a proxy-based interface specifically designed for Setonix to resolve domain names.

- Experiment process: Resolving was performed randomly without duplication for the top 100 prepared domain names. After resolving each of the 100 domains once in a single experiment, the cache was flushed. This process was repeated 20 times to obtain sufficient data.
- Data processing: The data obtained through the experiment process were aggregated for each item and divided by 20 to calculate the average. Finally, the result was divided by 100 to derive the average resolving time per attempt.

First, both Google and Cloudflare, the two conventional DNSs, showed an average latency of less than 50 ms for each resolving request. In contrast, the blockchain-based Handshake DNS showed a relatively higher latency, averaging over 200 ms. However, more significantly, in the case of Handshake, which uses a PoW-based consensus algorithm, it appears that the inherent TPS (Transactions Per Second) limitations of this algorithm have had a greater impact.

Additionally, for conventional DNS, there was no significant difference in latency between using the local resolver and passing through the proxy. However, a notable difference was observed in Handshake, which operates on a blockchain basis. This can be interpreted as the result of Handshake, which was created by a smaller community compared to established DNSs, being optimized for requests made through the more commonly used local resolvers rather than for those using open-sourced customized DNS tools. Additionally, the inherent lower performance of Handshake means that even relatively minor issues, such as an additional hop introduced by the proxy, can significantly increase latency.

Figure 7 is a graph comparing the resolving latency of two types of domain names in blockchain-based DNSs: Handshake and Setonix. The experiment was conducted in the same way as in Figure 6, using 100 domains of each DNS type, and the results show the average resolving latency for each individual request. All resolving tools used Setonix's proxy-based interface, and for Handshake, the internal custom resolver of the tool was pointed to the Handshake DNS.

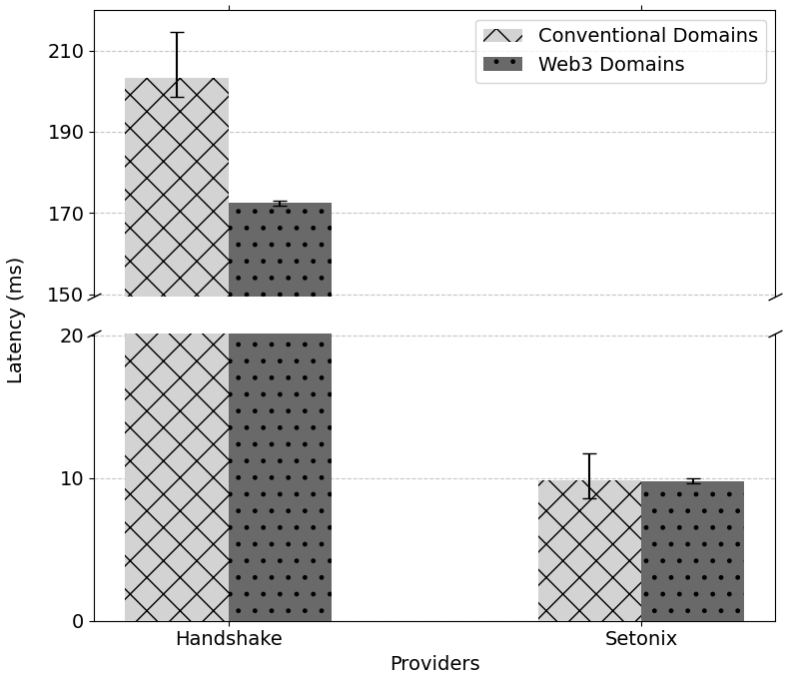

Figure 7. The difference in resolving latency for conventional domain names and internal Web3 domain names was measured for both Setonix and the blockchain-based DNS Handshake. Error bars indicate the maximum and minimum values for the average of each dataset.

The results show that Setonix is overwhelmingly superior to Handshake for both conventional and Web3 domain names. This aligns with the analysis of Figure 6, where Handshake's comparatively high latency was attributed to its operation on a PoW-based system with significantly slower TPS and its complex internal implementation. It can also be observed that Web3 domain names have over 15% lower latency compared to conventional domain names. This is because Web3 domains are uploaded to the internal blockchain and can be resolved immediately, whereas conventional domains must go through multiple steps, such as the TLD and SLD, via the internally implemented root zone. In contrast, Setonix shows no significant difference between Web3 domain names and conventional domain names due to its simpler internal implementation and the efficient domain classification by the protocol checker.

Figure 8 is a graph that allows a side-by-side comparison of the latency of Web3 domain names from three blockchain-based DNSs, ENS, Handshake, and Setonix, with two conventional DNSs. The experiment was conducted in the same manner as before, using a cache flush method. Since ENS does not allow resolution through resolvers that do not follow its Web3 query rules, the test was conducted with 100 pre-selected ENS domain names according to these rules. As with Setonix and Google, the data represent the average values.

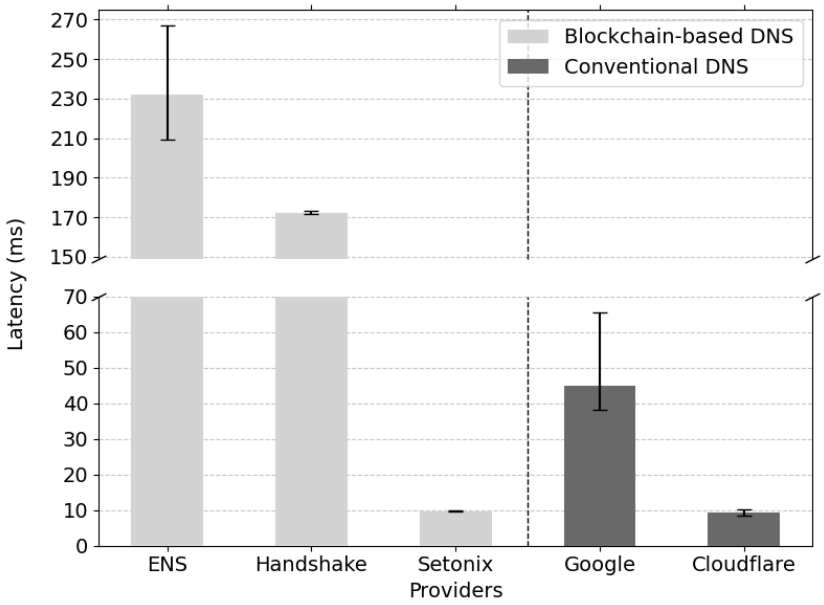

Figure 8. Resolving latency for each DNS: Web3 domain names for blockchain-based DNSs like 'superdomain.hdns' and conventional domain names for conventional DNSs like 'google.com'. Error bars indicate the maximum and minimum values for the average of each dataset.

As seen in the chart, ENS shows the worst latency, even slower than Handshake. This is because ENS has a similarly complex internal implementation as Handshake and also uses a more complicated method of exchanging data, relying solely on its Web3 query system instead of the standard DNS approach. Additionally, unlike conventional DNSs, which can optimize the path to the DNS server through methods like CDN or Anycast, ENS cannot use such methods. This further contributes to its lower performance. In contrast, Setonix shows latency comparable to Cloudflare even for its Web3 domains. This result is due to its simple internal components, efficient hierarchy processing methods, and the combination of a well-designed interface.

In Figure 9, caching is allowed, and to demonstrate the effects of caching, the graph shows how latency changes over successive attempts, unlike the previous experiment. Each time represents the result of resolving 100 different domain names, as in the previous experiment. The key difference here is that instead of showing the average values, the full

set of 100 attempts is displayed to illustrate the change in latency over time. Note that the vertical axis is in seconds. As can be seen from the legend in Figure 9, the dashed line represents the results of resolving through Setonix's proxy-based interface, while circles represent Google DNS, triangles represent Cloudflare, and squares represent Handshake.

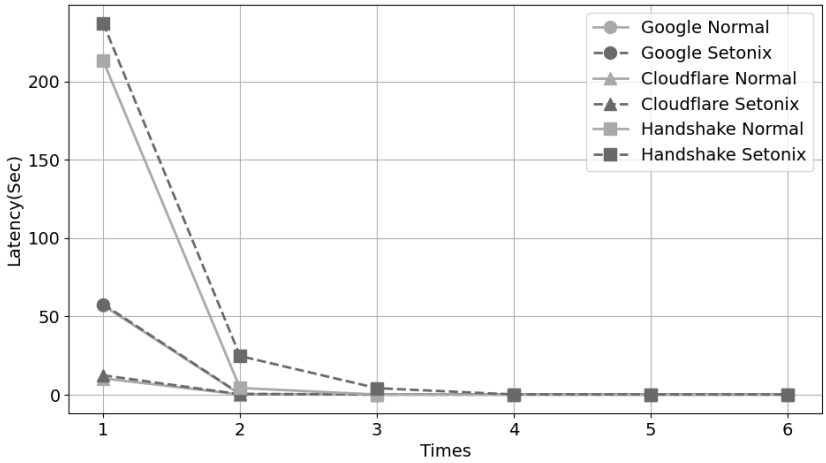

Figure 9. Results of performing six consecutive resolving attempts for each DNS without flushing the cache. In this figure, please note that the data for Google Normal (light circle symbol) overlaps with that of Google Setonix, making it visually indistinguishable. Therefore, the data for Google Normal can be considered nearly identical to that of Google Setonix when interpreting the graph.

The patterns seen with the two methods, local resolver and Setonix, are almost identical to the graph in Figure 6. What is particularly noteworthy is that, as expected, the resolving latency approaches zero as the number of attempts increases due to caching. Meanwhile, even with the algorithms currently in use, the internet usage of the average user is somewhat limited to a restricted area [29]. Through this, even when using a system with a relatively higher average resolving latency, the effect of caching becomes more pronounced due to the limited range of access typical for users. In the end, the combination of a cache-enabled proxy-based interface and Setonix, like the local resolver, is unlikely to cause significant latency-related inconveniences for users.

Figure 10 shows the Query Per Second (QPS) measurements for each DNS, using the DNS load analysis tool dnsperf [30]. The experiment utilized dnsperf on an AWS EC2 t3.medium instance (2 vCPUs, 4 GB RAM), running Ubuntu 22.04 in the Seoul region. Requests were routed through the Setonix proxy client on a PC with the specifications outlined in the first paragraph, also located in Korea.

What can be observed from the graph is that Handshake's higher latency is also reflected in its lower bandwidth, which could cause inconvenience for many users. In contrast, Setonix demonstrates over 10 times better performance, and while it does not match conventional DNSs, it offers much better usability compared to existing blockchain-based DNSs. Although the reduction in bandwidth caused by the custom resolver in the Setonix Interface may seem significant when looking at the graph, this interface is installed and used locally by each individual. Therefore, the bandwidth reduction does not impact the usability of the system for individual users.

Although we wanted to include experimental results for registration time, it was not feasible to measure it, as the registration process in each DNS involves multiple steps and can even include a waiting period of several days. However, with Setonix domains, registration is completed the moment a registration request is sent along with the necessary data for the contract, indicating that it can be considered to other DNS systems in terms of performance and convenience.

However, as mentioned at the beginning of the Experimental Results section, the testing environment is not based on Ethereum's main net but on the Sepolia test network.

Therefore, differences in results due to network scale should be taken into consideration. Additionally, the number of domains registered for testing was around 100, and the maximum number of concurrent users tested using dnsperf was limited to 100. Considering the scale of these tests, it is important to note that these numbers are far below what would be expected for an actual service environment. Nevertheless, blockchain-based systems inherently possess good load distribution performance due to their decentralized nature, allowing services to be provided across multiple nodes. This characteristic suggests that they are likely to exhibit performance not significantly different from the current results, even in large-scale scenarios.

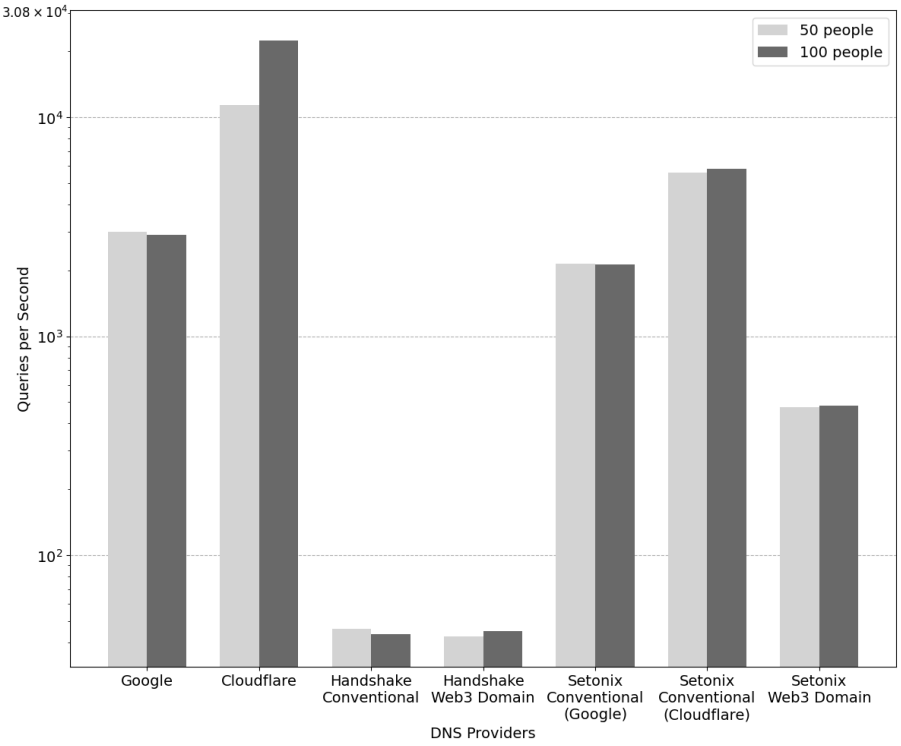

Figure 10. QPS for each DNS measured using the dnsperf program. Note that the y-axis representing QPS is on a log scale.

7. Related Works

In this section, we will explore attempts to resolve the issues of conventional DNSs using blockchain, as well as cases where blockchain has been utilized to support access to Web3 and blockchain-based DNSs. The study in [31] pointed out that while a conventional DNS supports security extensions through DNSSEC, it does not inherently support encryption, and it allows for unwanted webpages and services to be censored by authoritarian regimes.

To escape from such threats, alternative solutions for DNSs were explored, and one of the chosen options was blockchain-based DNSs. However, through actual experiments, it was demonstrated that blockchain-based DNSs also have potential for abuse, including the distribution and concealment of malware, exploitation of weak registration mechanisms, the permanence of malicious content and domains, as well as challenges related to system maintenance and trust. Based on these findings, this paper recognizes the vulnerabilities in blockchain-based DNS configurations and includes finding ways to improve them in future research. Ref. [32] proposes a solution to build a secure and reliable decentralized name service using blockchain technology to address the issues arising from the significant inherent design flaws of conventional DNSs. The advantages of blockchain-based DNSs include immutability, decentralization, and enhanced user authority. To explore this, the current state of blockchain-based name services is analyzed, focusing on ENSs.

ENS offers advantages such as the elimination of SPoF, guaranteed data immutability, and protection against various attacks like cache poisoning and DNS poisoning. However, it also has drawbacks, including the potential for abuse by attackers through squatting, malicious content, fraudulent addresses, as well as issues related to record permanence, complexity, and costs.

This paper has clarified the strengths to adopt and the areas for improvement in blockchain-based DNS design through the analysis of existing blockchain-based DNSs from various perspectives, as discussed in the referenced paper. In particular, even if a good system is built, it will not be used if its complexity makes it difficult for users to access it. Therefore, we focused on ensuring that the system is easy for users to use.

The study in [33] points out that conventional DNSs are vulnerable to both external attacks and internal threats due to their centralized structure, mentioning the 2002 incident in which 9 of the 13 root DNS servers were hit by a large-scale DDoS attack. It emphasizes not only external attacks such as DoS and reflection attacks but also the risks of internal attacks caused by malicious root servers. While the current DNS responds to external attacks through DNSSEC and BGP anycast, it still points out security vulnerabilities due to the potential for internal attacks. To address these issues, the paper proposes TD-root, which offers security through decentralized root-zone file management and the C-RAND consensus algorithm. This approach is expected to eliminate single points of failure (SPoF), prevent data tampering and censorship, and securely and efficiently protect the DNS. However, there is no mention of how users would actually use the system or generate their Web3 domains, which, like Handshake, makes it clear that TD-root's purpose is solely to replace the existing DNS, lacking broader objectives compared to Setonix. Ref. [34] points out that key components of the internet backbone, such as the Border Gateway Protocol (BGP), DNS, and Public Key Infrastructure (PKI), are vulnerable in terms of security, performance, and privacy due to their centralized architecture, and argues that blockchain can address these issues. The decentralization and security properties of blockchain can improve certificate transparency and revocation transparency in PKI, path validity in BGP, and the SPoF issue in DNSs. Ref. [35] provides a detailed discussion on DIDs, addressing the issues of traditional centralized identity verification systems. It also highlights performance problems caused by the network connection complexity inherent in existing DIDs and proposes solutions to address these challenges. DIDs are an excellent means of addressing issues such as privacy infringement and loss of data control inherent in traditional centralized identity verification systems. However, to resolve the usability challenges caused by low performance, attempts have been made to improve performance by incorporating Distributed Hash Tables (DHTs).

This approach achieved significant performance improvements. It demonstrated faster response times compared to traditional cloud-based systems and maintained stable throughput using DHT, in contrast to the sharp performance decline observed in cloud-based systems under increasing workloads. The inherent superiority of DHTs in terms of privacy compared to centralized identity verification systems could be applied to Setonix communications in the future. This would enhance security while offering a user experience at the same level or even better than before, thanks to the improved performance of the DHT-based approach.

Among these, DNSs, which are relevant to this paper, utilize distributed ledger technology to store DNS records on the blockchain, thereby eliminating SPoF, ensuring DNS data integrity, simplifying DNSSEC key management, and reducing response size to improve performance. However, like other approaches, there is no mention of supporting its Web3 domain names, unlike Setonix. This paper is more focused on solving the inherent problems of a conventional DNS, making its limitations clear.

In the overall body of research combining blockchain technology with DNSs, there have been more efforts to address the vulnerabilities of existing domains rather than focusing on Web3 support, custom domain names, or improving usability. This confirms that Setonix, as proposed in this paper, is a relatively new area of research within the broader research group. However, through other studies, several options were found to enhance

Setonix by adding security protocols, particularly by applying decentralized security techniques. This opens the path for improving the system in a more robust direction.

8. Conclusions

This paper raised the issue that blockchain-based decentralized Web3 services are using the centralized Web2 server–client model for service access. Due to this, the advantages of decentralized systems are lost, and vulnerabilities arise in the service because of the SPoF issue inherent in the centralized structure of the access path. To resolve this, it was necessary to introduce a decentralized access path, leading to the proposal of Setonix, which combines a decentralized DNS with a proxy-based interface for its use.

In conclusion, Setonix provides a decentralized access path option for Web3 services, enabling ownership verification without the involvement of central authorities through smart contracts. This results in a DNS that allows services to be uniquely and freely expressed. In addition, through the proxy-based interface, which efficiently categorizes domain names and enhances user experience, users can freely use both Web2- and Web3-based domains without needing any manual conversion after a one-time setup.

However, as it is implemented as a smart contract using Solidity, it has a clear limitation in that it can only operate in blockchain environments equipped with an EVM. Conducting future research to enhance its general applicability to typical blockchain networks would provide substantial value in supporting Web3. Additionally, future research will focus on exploring methods for directly connecting to nodes of services operating on blockchain p2p networks, including addressing deficiencies in security settings. In particular, incorporating DNSSEC extensions currently used in DNSs into blockchain systems or utilizing Decentralized Identifiers (DIDs) to enhance the security of blockchain-based DNSs could serve as valuable research topics.

In terms of scalability, further research is expected to overcome the current limitation of operating only on EVM-based systems, enabling functionality across various blockchain platforms. Such advancements could lead to a more versatile system, playing a significant role in positively promoting Web3 and other blockchain-based services.

Author Contributions: Conceptualization, J.J. and S.P.; Methodology, J.J.; Software, J.J.; Validation, J.J.; Investigation, J.J.; Resources, S.P.; Writing—original draft, J.J.; Writing—review & editing, J.J.; Visualization, J.J.; Supervision, S.P.; Project administration, S.P.; Funding acquisition, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Bisa Research Grant of Keimyung University in 2023 (Project No: 20230222).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Lai, Y. Web3: Exploring Decentralized Technologies and Applications for the Future of Empowerment and Ownership. *Blockchains* **2023**, *1*, 111–131. [CrossRef]
2. Korpal, G. Decentralization and Web3 Technologies. *TechRxiv* **2023**. [CrossRef]
3. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://static.upbitcare.com/931b8bfc-f0e0-4588-be6e-b98a27991df1.pdf (accessed on 23 November 2024).
4. Stephen, R. A Review on Blockchain Security. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *396*, 012030. [CrossRef]
5. 60 Best Crypto Gaming Blogs and Websites in 2024. 2024. Available online: https://coin.feedspot.com/crypto_gaming_blogs/ (accessed on 23 October 2024).
6. UniSwap Home Page. 2024. Available online: https://app.uniswap.org/ (accessed on 23 October 2024).
7. Jensen, J.R.; von Wachter, V.; Ross, O. An introduction to decentralized finance (defi). *Complex Syst. Inform. Model. Q.* **2021**, *26*, 46–54.
8. Opensea Homepage. 2024. Available online: https://opensea.io/ (accessed on 23 October 2024).

9. Filecoin Homepage. 2024. Available online: https://filecoin.io/ (accessed on 23 October 2024).
10. Web 3.0 Blockchain Market Size—By Type (Public, Private, Consortium, Hybrid), by Application (Cryptocurrency, Conversational AI, Data & Transaction Storage, Payments, Smart Contracts, Digital Identity), End-User Industry & Forecast 2023–2032. 2024. Available online: https://www.gminsights.com/industry-analysis/web-3-0-blockchain-market (accessed on 23 October 2024).
11. Banerjee, A.; Byrne, R.; De Bode, I.; Higginson, M. Web3 Beyond the Hype. 2022. Available online: https://www.mckinsey.com/industries/financial-services/our-insights/web3-beyond-the-hype (accessed on 23 October 2024).
12. Web 3.0 Market Size, Share, and Trends 2024 to 2034. 2024. Available online: https://www.precedenceresearch.com/web-3-0-market (accessed on 23 October 2024).
13. Shaikh, A.; Tewari, R.; Agrawal, M. On the Effectiveness of DNS-based Server Selection. In Proceedings of the Proceedings IEEE INFOCOM 2001: Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), Anchorage, AK, USA, 22–26 April 2001; Volume 3.
14. Hudaib, A.A.Z. DNS Advanced Attacks and Analysis. *Int. J. Comput. Sci. Secur.* **2014**, *8*, 63.
15. Alieyan, K. An Overview of DDoS Attacks Based on DNS. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 19–21 October 2016.
16. Sewak, M. Winning in the Era of Serverless Computing and Function as a Service. In Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018.
17. About Cloudflare DNS. Available online: https://developers.cloudflare.com/dns/ (accessed on 23 October 2024).
18. Handshake White Paper. Available online: https://handshake.org/files/handshake.txt (accessed on 23 October 2024).
19. Postel, J. *Domain Name System Structure and Delegation*; Technical Report, RFC1591; Internet Engineering Task Force (IETF): Fremont, CA, USA, 1994.
20. Mechant, P. An Illustrated Framework for the Analysis of Web2.0 Interactivity. *Contemp. Soc. Sci.* **2012**, *7*, 263–281. [CrossRef]
21. Ethereum Name Service Documentation. 2024. Available online: https://docs.ens.domains/web/quickstart (accessed on 23 October 2024).
22. About ICANN. Available online: https://www.icann.org/resources/pages/welcome-2012-02-25-en (accessed on 23 October 2024).
23. Von Arx, K.G. ICANN-Now and Then: ICANN's Reform and its Problems. *Duke Law Technol. Rev.* **2003**, *2*, 1.
24. Goerli Testnet Homepage. Available online: https://goerli.net/ (accessed on 20 November 2024).
25. About Sepolia Testnet. 2024. Available online: https://docs.etherscan.io/sepolia-etherscan (accessed on 23 October 2024).
26. Infura Homepage. 2024. Available online: https://www.infura.io/ (accessed on 23 October 2024).
27. Metamask Homepage. Available online: https://metamask.io/ (accessed on 23 October 2024).
28. Handshake DNS Domain Name Trading Site (Namecheap). Available online: https://www.namecheap.com/ (accessed on 20 November 2024).
29. Moeller, J.; Helberger, N. *Beyond the Filter Bubble: Concepts, Myths, Evidence and Issues for Future Debates*; University of Amsterdam: Amsterdam, The Netherlands , 2018.
30. Nominum. *dnsperf: DNS Performance Tool Manual*; Nominum: Redwood City, CA, USA, 2012.
31. Patsakis, C. Unravelling Ariadne's Thread: Exploring the Threats of Decentralised DNS. *IEEE Access* **2020**, *8*, 118559–118571. [CrossRef]
32. Xia, P.; Wang, H.; Yu, Z.; Liu, X.; Luo, X.; Xu, G. Ethereum Name Service: The Good, the Bad, and the Ugly. *arXiv* **2021**, arXiv:2104.05185.
33. He, G. TD-Root: A Trustworthy Decentralized DNS Root Management Architecture Based on Permissioned Blockchain. *Future Gener. Comput. Syst.* **2020**, *102*, 912–924. [CrossRef]
34. Ali, F.S. Improving PKI, BGP, and DNS using Blockchain: A Systematic Review. *arXiv* **2020**, arXiv:2001.00747.
35. Alizadeh, M.; Andersson, K.; Schelén, O. Comparative Analysis of Decentralized Identity Approaches. *IEEE Access* **2022**, *10*, 92273–92283. [CrossRef]