Blockchain Address Transparency with DNS

Mara Caldeira Miguel Correia
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Associação DNS.PT

***Abstract*—Blockchain systems have gained popularity in the last few years. The services they provide go from monetary transactions to digital forms of identification and many others. However, blockchain addresses are long numbers that are not adequate for human use, as they are hard to remember and manage. This work considers the use of the Domain Name System (DNS) as a means for users of any blockchain to use domain names to identify blockchain addresses. Similarly to the way DNS stores mappings of IP addresses to domain names, we offer a solution that allows it to also keep mappings of blockchain addresses to domain names. Our solution, *Domain Name System for Blockchain-type Addresses* (DNSBA), allows any blockchain address to be easily substituted by an associated domain name. Our solution provides two options in terms of DNS Resource Records and was implemented in BIND9. It was assessed in the context of a use case: a blockchain service for registering student diplomas. Its performance and cost are much better than the alternative solution of using a blockchain to store these mappings, as done by the Ethereum Name Service (ENS).**

I. Introduction

Since a few years ago, the deployment of blockchain applications has been growing at a fast-paced rate, making obvious the potential hidden in this technology [10], [14], [30], [32]. Nowadays, almost anyone can refer at least one or two times that the blockchain topic came into conversation in their daily lives. Despite the continuing interest in cryptocurrencies, blockchain is distancing itself from mere money transactions and investments, to applications more related to our lives than we have yet to realize.

However, *blockchain addresses are long numbers that are not adequate for human use*, as they are hard (if not impossible) to remember and manage. For example, a Bitcoin account may have the address `bc1qar0srrr7xfkfy5l643lydnw2re49gtdzrf8mnm`. Our main goal in this work is to provide a system that allows a blockchain user to identify his/her addresses in a more amenable way, e.g., `account.inesc-id.pt`. This problem is similar to the one observed for IP addresses several decades ago, so we use a similar solution, showing its benefits and shortcomings.

This work uses the *Domain Name System* (DNS) [15], [22] as a means to offer users – of any implementation of blockchain – to simply use domain names to identify blockchain addresses. Similarly to the way DNS stores associations of IP addresses to domain names, we offer a solution that allows it to also keep mappings of blockchain addresses to *domain names*. The DNS system is adequate for this use due to its resilience, i.e., to its high availability and integrity despite the severe level of threat that it is exposed to [13].

Our solution, *Domain Name System for Blockchain-type Addresses (DNSBA)*, offers the possibility of any blockchain address being easily translated into an associated domain name. Moreover, it leverages DNSSEC [7]–[9], [16] to provide additional security properties that are provided by most blockchain systems [6], [11], [18], [30], so also necessary in the system that provides translations of names to addresses.

DNSBA provides two options in terms of DNS *Resource Record* (RR) used: the TXT RR and a new RR we introduce (BC, for blockchain). We implemented this new RR in the classical and still one of the most used DNS clients, BIND9 [1], [26]. BIND can be used to publish *DNSSEC-signed* DNS root zones and Top-Level Domains (TLDs), to run a caching DNS server or an authoritative name server as well as to provide features such as: load balancing, dynamic updates, among others.

We assessed the benefits of DNSBA in the context of a use case: a blockchain service for registering student diplomas, developed in Project Qualichain [31]. We also evaluated experimentally the performance of DNSBA and provide a comparison with a competitor, the Ethereum Name Service (ENS) [6], that stores name-address mappings in the Ethereum blockchain. DNSBA has a much better performance than ENS (10s of seconds instead of many seconds of latency) and much lower cost (essentially free, instead of having every operation paid in ether).

II. Related Work

There are currently in place a few solutions to resolve a name into a blockchain address. However, they take in consideration a single blockchain-type address, or in the best case, a few chosen ones. Currently, a global solution like the one we propose in this document does not exist.

The first solution, for the Ethereum blockchain, is the *Ethereum Name Service* (ENS) [6]. This service is divided mainly in two components: *registrars* and *resolvers*. The first, a registrar, is a smart contract that owns TLDs like '*.eth*' or '*.test*', and that keeps the records of all domains and subdomains on the Ethereum blockchain. The later, a resolver, is a smart contract that has the same functions as a DNS resolver. A registrar specifies the rules of allocation of its subdomains, allowing anyone that wishes to obtain ownership of a domain for their own use, to do so by abiding such rules. This way, anyone who owns a domain may configure subdomains for themselves or others, as they see fit. Nevertheless, in Ethereum each smart contract has associated costs [3], therefore, every time a user uses ENS to register or resolve

a name, he/she has to pay an amount in ether (Ethereum's currency). A second solution is BitAlias [27]. BitAlias works very similarly to ENS, but it can only be used for Bitcoin addresses. Similarly to ENS, BitAlias also requires a fee for its usage (in bitcoins). The price, however, varies every six months and decided by BitAlias's holders. There are similar solutions such as OneName [12] and OpenName [19].

There is a larger literature on solutions to a problem that is sort of the opposite: using blockchain as a name service. The objective is to guarantee the security (authenticity, integrity) of the associations between names and addresses. *Blockstack* uses Bitcoin to provide a naming system that has no central point of trust and has a strong security [5]. *ConsortiumDNS* took the idea behind Blockstack and led it further in terms of security and performance, by designing a three-layer architecture and additional storage [33]. There are several others [20], [21], [25].

III. DNS for Blockchain-type Addresses

This section presents our solution.

A. Requirements

Our requirements are generically two. First, our solution cannot reduce the guarantees provided by blockchains. Second, it should be compatible with all blockchains. These general goals are translated into the following requirements:

- *Authenticity*: Each mapping is legitimate and not counterfeit;
- *Integrity*: Each mapping is accurate and can not be modified without authorization;
- *Availability*: The system is highly available and it is functional with average uptime;
- *Compatibility*: The system does not interfere with the normal functioning of DNS;
- *Universality*: The system supports all blockchain systems.

Notice these requirements cannot be entirely satisfied, i.e., guaranteed with probability 1. This is not necessary, as blockchains also do not guarantee their properties with probability 1. For example, it is known that the integrity of the Bitcoin chain of blocks may be violated by a large enough mining pool, although in practice this is unlikely and undesirable for the participants (nevertheless, it has happened with smaller blockchains like Bitcoin Gold and Ethereum Classic, not to be confused with Bitcoin and Ethereum).

B. DNSBA with the TXT Record

DNSBA aims to offer a service that enables the mapping of a blockchain address to its domain name through the use of the DNS (with DNSSEC). While envisioning how we could achieve it, we deliberated what would be a better approach, either the creation a new Resource Record made only for our purpose, or the usage of the existing TXT record. Since both approaches had their own advantages and downsides, we took the decision of considering both of them.

The *TXT record* allows a user to store any kind of data inside it, being format-free. Therefore, it would enable us to use it essentially as an *A record* and an *AAAA record*, only instead of maintaining a mapping of domain names to IP addresses, it could keep mappings between domain names of blockchain accounts and the multiple addresses that every account possesses.

```
mara@mbp16admin01 ~ % dig @127.0.0.1 txt.tecnico.ulisboa TXT

; <<>> DiG 9.16.8 <<>> @127.0.0.1 txt.tecnico.ulisboa TXT
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59112
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: dc5d0db6bbf5787601000000605e0031229f18f25b24bfd0 (good)
;; QUESTION SECTION:
;txt.tecnico.ulisboa.           IN      TXT

;; ANSWER SECTION:
txt.tecnico.ulisboa.    500     IN      TXT     "t=BTC f=1 0x2CefB619218825C0c670D8E77f7039e0693E1dDC"

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Mar 26 16:39:29 CET 2021
;; MSG SIZE  rcvd: 141
```

Fig. 1: TXT Record query

The TXT RR has no defined structure and that way, each application uses it as it sees fit, applying its own structure/syntax to it. This can lead to a difficulty in distinguishing one application's record from others. In fact, unless there is a prior agreement between the involved parties, that decided to use this kind of RR as a standardized protocol between them, it can lead to the problem of possible *collisions* with some other unsuited use of a TXT RR. One could try, nevertheless, to restrict the syntax that a certain application has for the TXT RRs and accept the possible hazards of such collisions, but doing so leads to possible unexpected issues of its own. nevertheless, RFC 5507 provides some guidelines on using the TXT RR [17].

The final concern we had in mind in this approach, was the matter of the space available for the payload of DNS messages. It could have been the case that the data produced would lead to larger messages than those permitted by the DNS design, making it impossible for our data to fit.

In our use case example (see Fig. 1), the possible format of the string of a TXT record can be separated into 3 main fields: *Type* ('`t=`'), *Format* ('`f=`') and *Value*. The *Type* defines the name of the type of blockchain address being used based on already known abbreviations, as mentioned above. The *Format* defines the format of the blockchain address, depending on the *Type* used, e.g. Bitcoin is represented as `BTC` and can have '`1´`, '`3`' or '`bc`' as format. Lastly, the *Value* defines the address itself.

The TXT record therefore allows for a single record to be composed of multiple strings which are then concatenated together without adding additional spaces, for example:

```
'txt.ulisboa TXT "t=BTC f=1 "
"1BvBSSEYsgWetqTAn5Au4m7GFg7xJaNVN2"'
```

C. DNSBA with the new BC Record

The second method that we decided to test as a possible implementation of DNSBA was to create a new RR type, specific for our purpose, allowing us to have no issues regarding possible misinterpretations of the data it contains.

```
mara@mbp16admin01 ~ % dig @127.0.0.1 tecnico.ulisboa BC

; <<>> DiG 9.16.8 <<>> @127.0.0.1 tecnico.ulisboa BC
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5050
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 83a79c1426fd3e8f01000000605dfdfeb5522e076acc7cec (good)
;; QUESTION SECTION:
;tecnico.ulisboa.               IN      BC

;; ANSWER SECTION:
tecnico.ulisboa.        500     IN      BC      "t=BTC f=1 0x2CefB619218825C0c670D8E77f7039e0693E1dDC"

;; Query time: 2 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Mar 26 16:30:06 CET 2021
;; MSG SIZE  rcvd: 137
```

Fig. 2: BC Record query

For our work, we decided that the best starting place was to analyze the TXT record implementation in BIND and replicate it for our own RR. Our thought process for this decision was the fact that the TXT record includes every functionality that we needed for our new RR, having as downside the fact that by using it we could potentially collide our implementation with another one of the same record. Therefore, by mimicking the TXT record structure, we could achieve our goal efficiently and without many (if any) emerging issues.

In terms of usability for a user, this new RR type should present no more effort to be used than the TXT record and therefore, at this point, it should only be considered a matter of preference from the user's perspective and of the system that benefits from our project.

We decided to name our new resource record *BC*, taking the first letter from each of the two words that constitute BlockChain. The BC RR is a DNS record that has as purpose to map domain names to blockchain addresses.

Notice that we did not go through the process of registering this new RR as this is a lengthy process and out of the scope of an academic work like this.

D. Using DNS to Translate Blockchain Names

In this section, we aim to explain how to use DNSBA to achieve our goal of translating domain names to blockchain addresses. To do so, we will explain how we used our custom RRs and how anyone can obtain the same results.

In order to run our own implementation of DNS, we needed to use a *daemon* called `named`, which is the executable obtained when BIND9 is compiled. A daemon, is a process that runs in the background and that answers requests for services. In this case, the `named` daemon is what permits DNS to be operational. After being enabled, `named` is invoked every time we perform a DNS request and the information found to answer our request is the one established in the DNS's configuration file `named.conf`.

To demonstrate our work, we added in the `named.conf` file a new custom made zone entitled *ulisboa* (Listing 1). This zone contains the specification to the file `test.ulisboa` that holds the RRs that belong to the said zone. It is therefore in the file `test.ulisboa` that we added our specially made RRs. Similarly to any other RR, we added our RRs by following the regular anatomy of a zone file (Listing 2).

Listing 1: zone definition in `named.conf`:

```
    zone "ulisboa" IN {
        type master;
        file "test.ulisboa";
        allow-update { none; };
    };
```

Listing 2: RR definition for `test.ulisboa`

```
$TTL    500 IN  SOA ulisboa. root.localhost.  (
                03261638   ; Serial
                28800      ; Refresh
                14400      ; Retry
                3600000    ; Expire
                86400 )    ; Minimum

@   IN  NS  localhost.
tecnico IN  BC  "t=BTC f=1 0
    x2CefB619218825C0c670D8E77f7039e0693E1dDC"
txt.tecnico IN  TXT "t=BTC f=1 0
    x2CefB619218825C0c670D8E77f7039e0693E1dDC"
```

What sets apart our solution from using regular DNS or its extension — DNSSEC — is the addition in the `named.conf` file `options` the commands:

- `dnssec-enable yes;`
- `dnssec-validation yes;`
- `dnssec-lookaside auto;`

After adding these DNSSEC options, we created a key pair specific for our purposes and we added once again in the `named.conf` file, a `control` clause (Listing 3) that states that if an access request originates from any other address that is not our own, they must provide the correct access key — `rndc-key`.

Listing 3: *DNSSEC keys* definition in `named.conf`

```
controls {
    inet 127.0.0.1 port 54 allow {localhost;};
    inet * allow {"rndc-users";} keys {"rndc-key";};
};
```

E. Properties of the Solution

This section discusses how our solution satisfies the requirements (cf. Section III-A).

Regarding *Authenticity*, we notice that DNSSEC assures the integrity of the RRs. More precisely, DNSSEC signs RRs and allows resolvers to check their integrity by verifying this signature. In terms of autenticity of the RRs inserted in DNS – TXT or BC –, we assume the owner of the domain assures only valid RRs are inserted.

Concerning *Integrity*, again this is ensured by the digital signatures provided by DNSSEC.

For the *Availability* requirement, our solution can assume to have the same average uptime functionality and availability as DNS itself. The DNS has a high availability, considering that apart from being necessary, it is frequently targeted by DDoS attacks. All in all, our changes do not modify DNS structure in some unfamiliar way since BIND9 allowed us to apply them

in an already preset and conventional form. Consequently, we can conclude that we effectively followed this precondition.

In terms of *Compatibility*, we simply use the TXT RR already provided by the DNS framework or the BC RR that is similar, so our solution does not interfere with the normal functioning of DNS.

Lastly, our solution at first sight satisfies the *Universality* as any address is representable in text and the TXT and BC records use text to represent them. A potential problem would be if the blockchain address the user wanted to support was too large for the data field in the resource record. RFC imposes some limits on the sizes of fields, e.g., 255 bytes for names [29], but not for the data fields of TXT RRs. Therefore, we have to assume that the limit that applies is a few bytes below 512 bytes to the whole RR that is the limit for a DNS message, although some environments may support more [17]. We believe it is unlikely that most address formats are larger than a few dozens of bytes (like Bitcoin's and Ethereum), much less hundreds.

IV. Evaluation

An important factor that we wish to note is the usefulness of our solution in applications that use blockchain. Our work therefore must not cause a big overhead in the regular operating time of the application, it has to create an actual improvement in the application between using our service and not. We also wish to assess the worth of our solution in comparison with similar solutions already in practice in the market, but for fewer blockchain-type addresses, such as *ENS*.

Our evaluation focuses on the *latency* of the solution, i.e., on the time it takes to resolve names. We do not consider throughput as the number of resolutions allowed by our system per unit of time is the one allowed by DNS; it makes no sense in measuring this throughput, that is not part of our work and not even static. Our main remaining question is: *Is it worthy it?* Our goal requires that DNSBA runs smoothly without any major increase in the request's processing time. If the increased delay is far superior to our expectations, our solution may either need further improvements or may not be worthy all together.

We therefore divide our evaluation in three scenarios.

- *Scenario 1*: What is the latency of DNSBA when an address is not in the cache, and when it is?
- *Scenario 2*: How much does an application's latency increase when it uses DNSBA versus when it does not?
- *Scenario 3*: How much lower is the latency of DNSBA in comparison to the latency of ENS?

A. Use Case: QualiChain Certificate Validation Ecosystem

We decided to demonstrate the utility of our work by applying its solution to project *QualiChain* [2], [23], [24], [28], [31], more specifically to one of its components, the *QualiChain Consortium Certificate Management Modules* [31]. *QualiChain* is a platform that mainly envisions to help organizations validate the educational qualifications and employment background of potential employees. The project explores the potential of blockchain technology by using it to register the certificates issued by proper *Certificate Authorities*, and allowing organizations to validate the authenticity and integrity of such certificates.

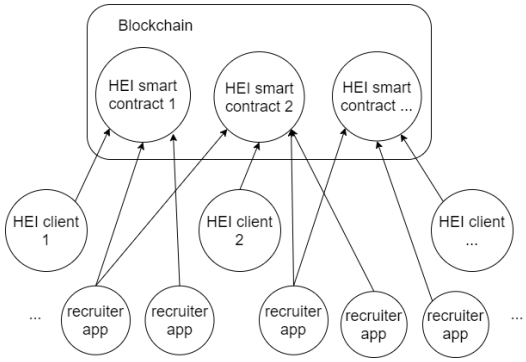

Fig. 3: QualiChain recruiter app data flow diagram

As previously mentioned, every blockchain implementation has its own use for blockchain addresses. In QualiChain, those addresses refer to smart contracts that are used to represent a *Higher-Education Institution* (HEI) in itself (i.e. HEI smart contracts), storing the data about the certificates issued by said HEI. This way, recruiters can run a *recruiter app* that allows them to validate if a certificate was indeed issued by a certain HEI (see Fig. 3).

In the current QualiChain application, a user interface requires the entity using the platform to insert a blockchain address that represents the certificate that they wish to validate.

Since the intent is for any organization to use this project to acquire certificates, we need to envision what issues a user not very familiar with technology would experience when faced with blockchain type of addresses. Therefore, turning QualiChain into an adequate use case for our work. Not only does our work improve the application usability for unfamiliar users, it also accelerates the process for those with enough insight on the matter.

The integration of our solution in this project allows, for instance, an organization seeking for the validation of a certificate that a certain individual — we shall name him João Silva — has issued by the HEI *Instituto Superior Técnico*, to simply type '`joao.silva.tecnico.ulisboa`' in the QualiChain's user interface. Essentially, instead of conducting a search through the entirety of the QualiChain network, the recruiter can solely go straight to the correct smart contract address, permitting him to quickly find the validation of the wanted certificate.

This implementation therefore succeeds in improving some time-consuming factors. It potentiates the recruitment process to go much faster, as the system of validation of qualifications is reduced to some quick search, similarly to a simple *DNS lookup*. Moreover, by using the DNS to perform address resolution, the operations performed to conduct the search for the smart contracts go much smoother, as each entity involved is already designed to execute this search on the network. All

of this while also using DNSSEC that aids in preventing cyber attacks against the integrity and authenticity of the system.

The expansion of QualiChain with our project resulted in a very simple but efficient solution. In the initial version, a user of QualiChain needed to insert the blockchain address of the certificate he/she wanted to authenticate in a field entitled as `IssuerID`. After our merge, whenever a user of QualiChain uses the application, in the field `IssuerID` he/she can simply type the domain name (e.g. `joao.silva.ulisboa`) associated with the individual whose certificate needs to be authenticated.

The main difference between the original QualiChain application and our extended version with DNSBA is an interception point that performs a DNS lookup, using the `dig` command for our costume made DNS (Listing 4).

Listing 4: DNS lookup in QualiChain Recruiting

```
function dnsLookup(hostname){
    return dig(['@176.111.104.55', hostname, '
        TXT']).then((result) => {
            var answer = result["answer"][0]["
                value"];
            var addr = answer.match(/"(.*?)"/)
                [1];
            return addr;
    }).catch((err) => {
        console.log('Error:', err);
        return err;
    });
}
```

B. Evaluation

To perform our evaluation scenarios in a realistic environment, we installed DNSBA in a remote server, provided by the Associação DNS.PT, the registry for the *.pt* domain. This server was created with the domain name `maracaldeira.devdns.pt` and with the IP address `176.111.104.55`. The main functionality of this server is to provide us faithful DNS resolver results, simulating how our solution would behave in a real-life environment.

1) Scenario 1: In our first scenario, as described above, we aim to evaluate the overall latency of our solution. With this evaluation, we wanted to find out how much time our work would take to resolve a mapping between a domain name and a blockchain address. As to acquire the most complete results, we created a script that performed a DNS query 1000 times in a row for a test domain that we created in our remote server.

Since most of the times a DNS query is executed without having the mapping already in its cache, we found interesting to divide the evaluation of our solution in two sub-scenarios, both performed under the same conditions:

- *Scenario 1.1 – cache flushed*: In this scenario, we implemented in the script a command for a DNS cache flush. This command is performed before every DNS query, therefore, the cache is always empty when the domain name to blockchain address mapping is required.
- *Scenario 1.2 – using cache*: In this scenario, we followed the normal procedure of a DNS query without changing anything. Therefore, in the first test query, the cache does not contain any value regarding our test domain but in the following 999 test queries, the cache is no longer empty and has saved the response of the previous query;

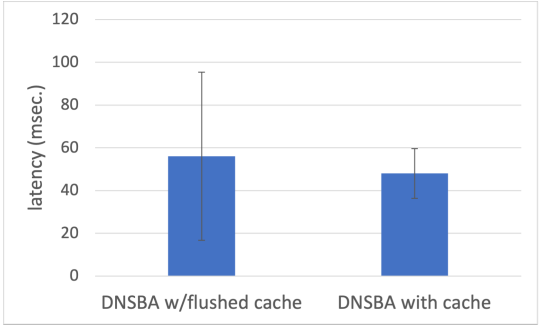

Fig. 4: DNSBA latency with flushed cache (Scenario 1.1) and with cache (Scenario 1.2)

By observing Fig. 4 we can conclude that if the cache is loaded, the regular query time should be around 48 msec with a possible deviation of 11.6 msec. On the other hand, if the cache is flushed, it should take almost 10 msec more to perform the query and it has a standard deviation far superior of that with cache, assuming a value of 39.3 msec.

When analyzing the results gathered, we can perceive that their difference is not very significant. We could argue that this may be due to the server and the client being very close to each other and/or in the same *Local Area Network* (LAN), however, in our tests the server was located in Portugal and the client in Switzerland.

Considering the information acquired by the statistics, we can declare that our solution does not cause any overhead in the regular DNS query performance, as the values observed are negligible. We can also conclude that even if the cache is empty for a request, the increase in the value of the query time is not considerable enough to be a concern. We can therefore determine that our solution passed our settled goals and that the latency increase of DNSBA is not greater than forecasted.

2) Scenario 2: In our second scenario, we take the regular QualiChain environment and compare it to our upgraded version — using DNSBA. This scenario has the particular issue of the fact that we need to take in consideration that our solution will obviously increase the time estimated for the QualiChain application to run, since it adds one more step in its procedure.

Similar to before, we decided to divide this scenario into two others. Since our concern is the total time spent by a user in the QualiChain application, to validate a certificate, we determined that the two scenarios would be the following:

- *Scenario 2.1 – QualiChain with DNSBA*: In this first scenario, we registered the time that QualiChain took to answer an user when using DNSBA to map the name domain inputted by the user, to the blockchain address associated with the said name domain. Once again, this scenario has as foundation the fact that the user has

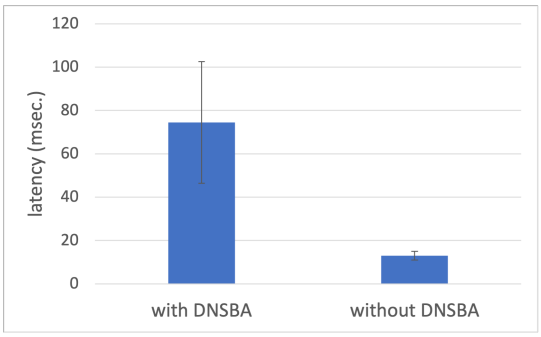

Fig. 5: QualiChain latency comparison with DNSBA (Scenario 2.1) and without DNSBA (Scenario 2.2)

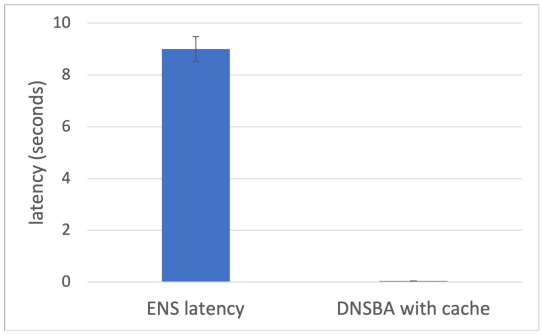

Fig. 6: Latency comparison between ENS and DNSBA

already inserted the input (in this scenario, the domain name) prior to the testing.

- *Scenario 2.2 – regular QualiChain*: In this second scenario, we took the regular QualiChain — which has as input from the user a blockchain address — and registered the time the application took to validate the certificate as of the time the user had finished typing the input.

Similarly to Scenario 1, we decided to evaluate our results by applying a median and standard deviation calculus to them. The results of this scenario were taken by changing the QualiChain application to run 1000 times the whole validation process. These tests were once again applied using the remote server in our possession to perform the DNS query, however the QualiChain application was set locally.

In Fig. 5 we can observe the results of our analysis. By examining the results, we can understand that when using our solution, QualiChain takes in median 74.5 msec to complete the validation process with a standard deviation of 28.1 msec. On the other hand, the regular QualiChain environment will validate the certificate in a median time of 13 msec with a standard deviation of 1.99 msec. As anticipated, the time taken by QualiChain to validate a certificate is superior when using DNSBA versus when performing its regular procedure. Nevertheless, these results where anticipated since we knew we were adding an extra procedure to the process.

We can therefore conclude that the time required to perform the validation of a certificate, even though it is increased while using DNSBA, it is still acceptable. Having in mind that our solution can be performed in 74.5 msec we believe that the application does not suffer a big increment in its process time. Moreover, since it is an operation that will not be performed many times in a row and, a delay under 100 msec is considered very small to be perceived by human sensitivity, our results are excellent. Thus, the results show us that QualiChain largely benefits from using DNSBA, as it can make its users more prone to use it more regularly since they would no longer have the inconvenience of typing the large addresses that blockchain has.

3) Scenario 3: In our last scenario we compare DNSBA's latency to ENS's — a similar solution to ours. In this last scenario we aim to determine either the benefit of DNSBA's overall query time in comparison to other solutions or to formulate some conclusions regarding the potential of our solution, even if its query time is far superior to ENS's.

In this final scenario, we analyze how much time does it take for DNSBA to resolve a mapping of a domain name to a blockchain address, versus the same process in ENS. We remind that we had already performed tests in Scenario 1 regarding our solution's latency, thus, in this scenario we simply compare the same results to those performed by ENS. Once again, we saw it as beneficial for our study to perform an evaluation based on two scenarios:

- *Scenario 3.1 – ENS*: In this scenario, we respect ENS's regular procedure and allow it to retain its cache data — after performing the first query — similarly to Scenario 3.1, for the remaining 999 tests queries.
- *Scenario 3.2 – DNSBA*: In this scenario, we apply once again the same scenario of that described in Scenario 1.2. We perform a regular DNS query and save in cache its results, leading the following 999 query tests to be deployed on DNS while retaining the previous query in cache.

In [4], we can observe every block created in Ethereum. By selecting each block individually, we can analyze the `timestamp` of their creation. When comparing the time difference between each creation, we can collect the 1000 values that allowed us to perform our evaluation of ENS.

By analyzing every timestamp gathered, we deduced that the time it took between each block creation was very close to the time that a new transaction would need to be performed. Thus, by compiling the results, as seen in Fig. 6, we obtained a median performance of 9 seconds with a standard deviation of 12.87 seconds.

By comparing the results found in Scenario 1.1 (Fig. 4) and the ones gathered in Scenario 3.2 (Fig. 4) we can see an astonishing difference in performances (Fig. 6). The Scenario 1.1 showed us that DNSBA accomplished a median time of 48 milliseconds for completing a query, while on the other hand, the Scenario 3.2 exhibits a median transaction time of 9 seconds. We can therefore evidently observe the value of our solution to any blockchain application that requires a quick and efficient solution.

V. Conclusions

This document presents the *Domain Name System for Blockchain-type Addresses* (DNSBA), a blockchain name system that provides the service of mapping domain names into blockchain addresses. Through DNSBA, we grant access for any type of application to simply connect to DNS and query for a blockchain address by using a simple mnemonic domain name. Apart from being a mere translator, DNSBA is also interesting since it uses DNS, taking advantage of its benefits and security functions. This mere factor permits a stronger security on the already very secure blockchain and it permits a new set of systems to be created on it.

Acknowledgments

This work was supported by Associação DNS.PT, by the European Commission program H2020 under the grant agreement 822404 (project QualiChain), and by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID). We warmly thank Dr. Marta Moreira Dias and Eng. Eduardo Duarte for their support to the project, and Prof. Sérgio Guerreiro and Prof. Rui Cruz for comments on earlier versions of this work.

References

[1] BIND 9, May 2020. https://www.isc.org/bind.

[2] QualiChain, May 2020. https://github.com/QualiChain/consortium.

[3] Ethereum Name Service: Registrar Frequently Asked Questions, 2021.

[4] Etherscan: Blocks. *https://etherscan.io/blocks*, May, 2021.

[5] M. Ali, J. Nelson, R. Shea, and M. J. Freedman. Blockstack: A global naming and storage system secured by blockchains. In *2016 USENIX Annual Technical Conference*, pages 181–194, 2016.

[6] A. M. Antonopoulos and G. Wood. *Mastering Ethereum: building smart contracts and dapps*. O'Reilly Media, 2018.

[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. IETF RFC 4033, March 2005.

[8] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol modifications for the DNS security extensions. IETF RFC 4035, March 2005.

[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource records for the DNS security extensions. IETF RFC 4034, March 2005.

[10] R. Beck. Beyond Bitcoin: The rise of blockchain world. *Computer*, 51(2):54–58, 2018.

[11] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. In *2015 IEEE Symposium on Security and Privacy*, pages 104–121, 2015.

[12] CH&Co. Onename gives users a public profile and username to register and manage their blockchain ID. *https://fintank.chappuishalder.com/case-studies/onename-blockchain-id/*, May, 2021.

[13] D. Conrad. Towards improving DNS security, stability, and resiliency. *Internet Society*, 2012.

[14] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang. Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385, 2018.

[15] D. Eastlake, E. Brunner-Williams, and B. Manning. Domain Name System (DNS) IANA considerations. IETF RFC 6895, September 2000.

[16] D. Eastlake and C. Kaufman. Domain name system security extensions. IETF RFC 2535, March 1999.

[17] P. Faltstrom, R. Austein, and P. Koch. Design Choices When Expanding the DNS. IETF RFC 5507, April 2009.

[18] A. E. Gencer, S. Basu, I. Eyal, R. Van Renesse, and E. G. Sirer. Decentralization in bitcoin and ethereum networks. In *International Conference on Financial Cryptography and Data Security*, pages 439–457. Springer, 2018.

[19] S. Gooding. New Plugin Adds Openname Avatars to WordPress. *https://wptavern.com/new-plugin-adds-openname-avatars-to-wordpress*, May, 2021.

[20] S. Gourley and H. Tewari. Blockchain backed DNSSEC. In *International Conference on Business Information Systems*, pages 173–184. Springer, 2018.

[21] Z. Guan, A. Garba, A. Li, Z. Chen, and N. Kaaniche. Authledger: A novel blockchain-based domain name authentication scheme. In *5th International Conference on Information Systems Security and Privacy*, pages 345–352, 2019.

[22] P. Hoffman, A. Sullivan, and K. Fujiwara. DNS terminology. IETF RFC 7719, Dec. 2015.

[23] I. R. Keck, M.-E. Vidal, and L. Heller. Digital transformation of education credential processes and life cycles: A structured overview on main challenges and research questions. In *12th International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2020)*, 2020.

[24] C. Kontzinos, O. Markaki, P. Kokkinakos, V. Karakolis, S. Skalidakis, and J. Psarras. University process optimisation through smart curriculum design and blockchain-based student accreditation. In *Proceedings of 18th International Conference on WWW/Internet*, 2019.

[25] P. Lai. Why DNS on Blockchain is the next step after DNS over HTTPS. *https://diode.io/distributed-infrastructure/Why-DNS-on-Blockchain-is-the-next-step-after-DNS-over-HTTPS-19231/*, May, 2021.

[26] C. Liu and P. Albitz. *DNS and BIND*. O'Reilly Media, Inc., 2006.

[27] Y. Malahov. BitAlias 1, aka usernames for Bitcoin, a new, simple naming system for Bitcoin addresses. *https://medium.com/bitalias-decentralized-naming-and-identity-service/bitalias-7b66bffed9d8*, 12, 2015.

[28] A. Mikroyannidis, A. Third, and J. Domingue. Decentralising online education using blockchain technology. In *The Online, Open and Flexible Higher Education Conference: Blended and online education within European university networks*, Oct. 2019.

[29] P. Mockapetris. Domain names—implementation and specification. IETF RFC 1035, Nov. 1987.

[30] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system," http://bitcoin. org/bitcoin. pdf. 2008.

[31] D. Serranito, A. Vasconcelos, S. Guerreiro, and M. Correia. Blockchain ecosystem for verifiable qualifications. In *2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pages 192–199, September 2020.

[32] S. Underwood. Blockchain beyond Bitcoin. *Communications of the ACM*, 59(11):15–17, 2016.

[33] X. Wang, K. Li, H. Li, Y. Li, and Z. Liang. ConsortiumDNS: A distributed domain name service based on consortium chain. In *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)*, pages 617–620, 2017.