DigiByte Domains: A DigiByte Name System

Renzo Diaz
renzo.diaz@remadi.net
digibytedomains.com

Abstract. A public association record between domain names and addresses that allows DigiByte blockchain users to send DigiByte and DigiAssets without the need of remembering the other party address, without extra cost. This proposal includes the usage of unlocked and locked DigiAssets to allow tagging and tracking of DigiByte addresses. Wallets can convert a DigiByte domain to a DigiByte address by querying an internal record without trusting a third party. For the convenience of a light implementation, a trusting mechanism is also provided.

1. Introduction

Operations between individuals in the cryptocurrency space always requires a high level of attention to detail and caution when sending and receiving funds. The design of DigiByte addresses, inherited from Bitcoin that uses *bech32* or *base58* encoding [1], makes them inconveniently hard for humans to remember them. This can lead to errors causing lost funds due to malpractice or scams.

For example, a common practice when sending DigiByte or DigiAssets is to use the machine's clipboard to *copy-paste* the desired address into a wallet sending form. This behavior, if the computer has been compromised by a virus that changes the original address, can lead to the fund's loss. Usually, this practice is accompanied by checking address's first or last characters, which can help by preventing fraud with day-to-day addresses that are generated at almost the same time, but for public addresses, published by organizations or companies, fake ones with a similar structure can be generated in a relatively short period of time.

What is needed is an integrated environment that allows the usage of easy to read and standard strings (domain names) to replace the hard to remember DigiByte addresses face to the user. This environment must provide some basic functionalities to transact with domain names. Some of these are: mechanisms to control the issuance of domains to prevent malicious individuals to hoard keywords; allow users to manage their domains to update the addresses that these represent; burn the no longer required domains; commerce with the domains they own; and private conversion between domain names to real addresses to create transactions and do not require a third party for this process. Under all circumstances trust must not be sacrificed.

All these functionalities can be achieved by creating a virtual implementation of a DNS (Domain Name System) using one of the pillars of the DigiByte ecosystem: DigiAssets.

2. DigiAssets

DigiAssets are small pieces of information attached to a transaction's output. They are virtual representations of things in the real and virtual world and can be issued, transferred, and burned using the same rules that imply creating a DigiByte transaction [2]. With the last implementation of DigiAssets v3 the environment has become more decentralized [3], which allows the creation of more projects on top of this platform.

Because of the convenience of an already built platform, and a community that supports the project, the proposal has been designed to be an implementation on top of the DigiAsset protocol.

3. Architecture

The DigiByte Name System architecture is like the well-known www domain names, and it consists of two main parts: the Domain DigiAssets and the DNS DigiAsset. Each of these are essential in the environment the DigiByte domains creates.

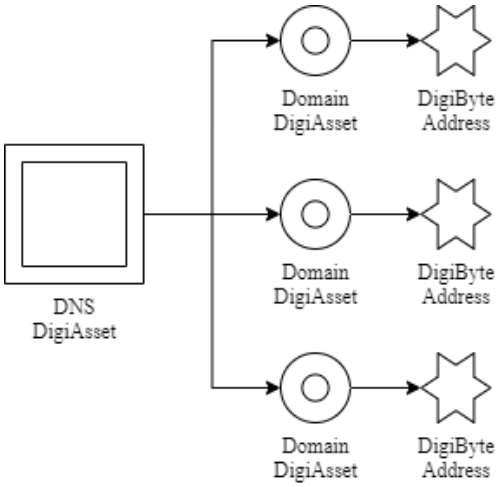

Figure 1

A Domain DigiAsset (*Domain*) is a non-fungible token on top of the DigiByte blockchain. There must only exist one unit of it per name, must be a locked, aggregable and with no divisibility DigiAsset [4] and must contain the title of the domain name which represents. By themselves, the *Domains* are indistinguishable from the regular DigiAssets, they need a central database to coordinate them.

The DNS DigiAsset (*DNS*) is a non-fungible token with no countable units, that contains on its metadata the list of all the names in circulation matched with their corresponded DigiAsset ID. It creates a traceable relationship between names and *Domains* and must be an unlocked, aggregable and with no divisibility DigiAsset [4].

The *Figure 1* shows how the *DNS* reference all the *Domains* in circulation and how each *Domain* reference only one DigiByte address. Is impossible for a *Domain* to reference more than one address due to the properties of the DigiAsset. As there is only one divisible unit, only one address can hold that *Domain* at a given time, but one address can be referenced by different *Domains* at the same time.

4. Characteristics

A *Domain* name is a plain-text string divided in two parts, the body, and the suffix.

The body's length must be between three and sixty-four-characters, and it is composed exclusively of lowercase characters and hyphens, but it can't start or end with a hyphen.

The suffix is standard for all the domains and must be ".dgb" and helps wallets to identify a *Domain.* It can't be changed, but the system supports new types of suffixes to be added in the future.

The *Domain* names are not case sensitive, when encoded and query in the *DNS*, the original string must be transformed to lowercase characters first.

5. Issuance

The system is meant to be partially centralized. The issuance of new *Domains* is in control of the key-masters and only they are allowed to issue new names under their established rules to avoid possible violation of trademarks and impersonation scams. Each time a *Domain* is created by the key-masters, they must also issue a new version of the *DNS* and its metadata must include the new name. It is imperative that every issuance is an exact update of the last one plus the new Domain and no DigiAsset ID is changed with another one.

The *DNS* can be audited by anyone by checking the list of versions on IPFS and for any X version of the asset all the name-id records on the X-1 version must be the same or no-existent with the only exception of a domain censorship. If inconsistencies are found, the *DNS* will be invalidated, and a consensus will have to be reached to issue a new one.

A domain censorship occurs when a trademark has been infringed with the purpose of scam or confuse the public. To execute a censorship the entry of the malicious domain must be changed to blank in the *DNS*. This process will invalidate the domain and it cannot be issued again.

6. Transfer and burn

As the DigiByte domains rely on the DigiAsset protocol, the transfer of domains between addresses or the decision of burning follows the same rules [4] as regular DigiAssets and can be managed by any wallet that supports this DigiByte feature. This also means that the domain can be in danger of an accidental burn if the UTXO where it is encoded is unintendedly used in a non-asset transaction.

7. Privacy

The pseudo anonymity of UTXO blockchains like DigiByte [2] create a standard in the crypto space and the DigiByte domains are intended to facilitate the public interaction between individuals, so they are not an option for people who looks for extreme privacy.

The DigiByte domains are compatible with the *do-not-reuse-addresses* principle. The process starts with a brand-new DigiByte address where an individual sent a domain to tag it. Then, that address starts receiving funds in multiple transactions. And finally, the owner creates a DigiAsset transaction that both transfers the domain to a new address and sends the remaining funds to another one.

8. Integration

The integration of the DigiByte Domains into a wallet relies on two modules. Each of them is independent from the other and can be developed or integrated separately into the desired wallet.

The first one is the sending module which implements the *DNS* and allows a wallet to send funds to any DigiByte Domain. The second one is the DigiAsset module and implements all the features to manage DigiAssets. In this document only the first module will be described as the second one is general purpose for all DigiAssets.

All sources must be trustworthy. For personal sources, it is recommended to have a local copy of the DigiByte blockchain for extra security. In case a third-party source for light wallets is chosen, this service must be public and trusted by the community.

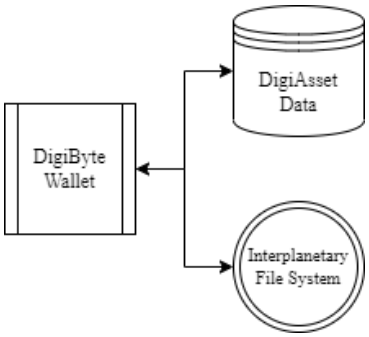

Figure 2

The *Figure 2* shows the two main sources a wallet must query to implement the sending module. The first one is the DigiAsset Data (*DAD*), this is a repository where all the DigiAsset information is parsed for easy access. The second one is the Interplanetary File System (*IPFS*) which is the off-chain storage used by the DigiAsset v3 protocol to store the DigiAssets Metadata.

The process to convert a DigiByte Domain into a DigiByte address can be summarize in 5 steps if the flow isn't interrupted.

a) Query the *DNS* id in the *DAD*.
b) Obtain the last hash of the *DNS IPFS* metadata.
c) Query the *IPFS* hash.
d) Lookup for the *Domain* AssetID in the metadata.
e) Query the AssetID in the *DAD* and look up for the unique holder.

On steps ***d*** and ***e*** there are some possible errors that the process might throw and must be in consideration to mark a conversion as invalid:

- If the *Domain* has not been claimed there is not going to be an entry.
- If the *Domain* has been censored, there is going to be an empty entry.
- If the *Domain* has been burned, there is not going to be a holder.

A wallet is not required to have integrated DigiAssets to integrate DigiByte domains. This leads to an easy adoption as the implementation process requires the development of only a few requests to a secure data storage for light wallets. For full wallets it may require some extra work by having the complete blockchain scanned to have the DigiAsset Data, but it can be forked from the DigiAssets implementation.

9. Conclusion

It has been proposed an architecture based on the DigiAsset protocol that allows the naming of DigiByte addresses. This architecture relies on a DNS DigiAsset that stores the relationship between domain names and DigiAsset ID. To ensure the fair issuance and the protection of trademarks a central authority must hold the keys of the DNS DigiAsset, but every user oversees their own domain represented by its unique DigiAsset. This system facilitates the sharing of DigiByte addresses between parties and can be implemented by querying any trusted data sources that contains the DigiAssets information.

10. References

[1] DigiByte (2020). *DigiByte Integration Guide v1.6.* https://digibyte.org/docs/integrationguide.pdf

[2] DigiByte (2019). *DigiByte Community Infopaper v1.0.* https://digibyte.org/docs/infopaper.pdf

[3] DigiAssetX (2022). *About digiassetX.* https://digiassetx.com/

[4] DigiAssetX (2021). *'DigiAsset' DigiByte Transactions.* https://github.com/digiassetX/DigiAssets-Protocol-Specifications/blob/master/DigiAsset-Scheme.md