

# Refining Ethereum Name Service: Constructive Improvements with UNS

Summary

Yamen Merhi

August 10, 2024

# Abstract

The Ethereum Name Service (ENS) has played a crucial role in the development of decentralized naming services, allowing users to associate human-readable names with blockchain addresses instead of long hexadecimal strings. While ENS has integrated seamlessly with the Ethereum blockchain, some design choices can be improved to create an even more effective naming system. The Universal Name System (UNS) builds upon ENS's foundation, addressing these limitations through targeted improvements.

This paper offers a concise summary of a more comprehensive research study. It examines key challenges ENS faces, such as name fairness assignment, record persistence, and user experience challenges, and introduces UNS as a refined solution. The summary aims to provide an overview of the insights and advancements that UNS offers, enhancing the foundational work done by ENS.

## Introduction

Since its launch in 2017, the Ethereum Name Service (ENS) has undergone various structural changes and employed different methods for assigning and managing names and subnames. The current architecture is represented by several key components:

- **ENS Registry:**
  - Stores ownership information for each name.
  - Stores the resolver address that contains information about each name.
  - Stores the TTL (Time to Live) for each name.
  - Allows the name owner to modify name configurations.
  - Allows the name owner to assign subnames.
- **ENS Registrars:**
  - The most well-known is the **EthRegistrar**:
    - Controllers can register names according to different logic.
    - **.eth** names are ERC721 NFTs that can be traded.
    - Different controllers are available, that can register names by paying a specific price based on the length of the name and its availability.
- **ENS Resolvers:**
  - Stores information on the names with a standardized interface and functions.

Additionally, there are other contracts associated with ENS that are beyond the scope of this research.

Despite the effective hierarchical structure made by ENS, certain issues persist within some of these contracts. This paper will discuss these problems and propose solutions to address them, aiming to enhance the overall functionality and user experience of the Ethereum Name Service.

# ENS Limitations

## 1. Record Persistence

The record persistence issue arises when the records of a name continue to be presented and resolved by wallets and interfaces, even if the name is no longer valid according to the registrar responsible for assigning it.

For instance, `.eth` names operate under a specific logic defined by the `.eth` registrar. As long as these names are registered and the yearly registration fees are paid, the owner retains control over them. However, if the owner fails to renew a name, such as `0xYamen.eth`, the records associated with this name—like address, content hash, and other properties—will still resolve, even though the name is no longer owned by "Yamen." This will remain the case until someone else registers the name and either resets the records or updates them with new information. The same applies to any subnames, like `vault.0xYamen.eth` or `treasury.0xYamen.eth`. These subnames, relevant to the previous owner, will still resolve, and the new owner might inherit them without being aware, with the subnames continuing to resolve based on outdated information.

This issue is not limited to `.eth` names. The problem is not with the `EthRegistrar` itself but with the absence of a standardized way to determine whether a name remains valid according to its registrar.

For example:

- **`vault.0xYamen.eth`**: Is the subname "vault" still valid under `0xYamen.eth`? (This might depend on custom logic).
- **`0xYamen.eth`**: Is `0xYamen.eth` still valid under `.eth`? (Has it expired, and have the registration fees been paid?).

This issue can have negative implications in two significant cases. First, for users who register names that may not be of interest to others, they could pay the registration fee once, and set up their records, and even if they don't renew the name, it can still be resolved. This leads to a loss of revenue for the `.eth` registrar, as names that should require ongoing fees continue to function without them.

Second, this could be dangerous for users, as it creates false assumptions about the validity of the information tied to a name. If a malicious interface detects the use of such a name, it could trigger a transaction to re-register the name. Once re-registered, the new owner could update the resolved information, potentially leading to harmful consequences, especially in cases involving the transfer of value. Users might unknowingly send funds to the wrong address based on outdated or maliciously altered records.

This issue is evident in cases like [thisisme.eth](#), where some subnames are still valid and are being resolved by wallets, even though the main name might have expired.

## **2. Name Fairness Assignment**

One of the significant issues with ENS is the fairness of name assignment, particularly concerning the low cost of registration. Today, ENS names are very cheap to acquire—one can register a name for as little as \$5 per year. While this seems reasonable for individuals registering a single name, it becomes problematic on a larger scale. At the launch of any naming system, especially one as popular as ENS, individuals or entities can buy up a large number of names at minimal cost, anticipating significant profit by reselling them on secondary markets. This is particularly true for high-value names, such as brand names, where the original registration fee is a fraction of what buyers are willing to pay later.

The problem is twofold. Firstly, the `EthRegistrarController` currently treats all names equally, without considering their potential value. A five-letter name that is only of interest to one person is treated the same as a five-letter name that represents a well-known brand, which many would be eager to acquire. This creates an unfair advantage for early registrants who can afford to buy these names cheaply and hold onto them indefinitely, waiting for a lucrative opportunity to sell.

Secondly, the registrar allows these names to be registered for extended periods—potentially even over a hundred years—by simply paying the current registration fees upfront. This practice locks in the ownership of names for long durations, potentially forever, without accounting for changes in the value of the names or the economic environment. Unlike property taxes, which are assessed and paid annually, allowing adjustments for inflation or changes in property value, ENS's current system does not accommodate such flexibility. As a result, even if future adjustments to registration fees or policies are made, they cannot affect names that have already been prepaid for decades.

This approach to name assignment can lead to significant issues, including reduced revenue for ENS DAO, as the low initial fees do not reflect the actual value of the names being registered. It also undermines the principle of fairness, as those who were early to register have an outsized advantage over new users who face much higher prices in secondary markets. Moreover, the rigidity of the system means that any future changes to how names are valued or managed cannot be retroactively applied, potentially locking the system into outdated pricing and allocation models.

## **3. Voluntary Release of Ownership**

A third issue with the current ENS system is the inability for users to voluntarily release ownership of a [.eth](#) name and make it available on the market again. Once a name is

registered for a specific period, the only option available is to transfer ownership to another party. There is no mechanism for a user to effectively "burn" their ownership and return the name to the pool of available names.

This limitation can be problematic in scenarios where a name is no longer relevant to the owner, and they wish to dissociate themselves from it. The lack of an option to release the name back to the market not only restricts user control but also has broader implications. It reduces potential revenue for ENS DAO, as names that could be of interest to new users remain off the market. Moreover, it perpetuates an unfair system of name assignment, as names registered under old rules continue to be held, without aligning with any new policies or pricing structures that the DAO might implement in the future.

#### **4. UX Limitations**

The current ENS system presents certain user experience (UX) limitations that could be improved for greater efficiency and security.

One limitation is found in the ENS Registry itself. When a user registers a name, there is no option to set the resolver data in the same transaction. This means the user must complete two separate transactions: one to register the name and another to set the resolver data. This process not only requires additional steps but also incurs additional transaction fees and time, which can be inconvenient for users.

Another UX limitation exists within the [.eth](#) registrar. Since [.eth](#) names are represented as NFTs, transferring the ownership of the NFT does not automatically trigger the transfer of the name in the ENS Registry. As a result, even if the [.eth](#) name NFT is now owned by a new entity, the ownership is not fully recognized until a second transaction is made to update the registry through the [.eth](#) registrar. This two-step process can lead to significant cumulative costs in transaction fees, especially considering the number of [.eth](#) names that have been transferred since the system's launch. These additional costs could have been avoided if ownership were directly updated in both places upon transfer.

## **UNS Improvements**

### **1. Addressing Record Persistence**

The issue of record persistence in ENS cannot be resolved by a single approach or entity alone, but by implementing rules that, if adhered to by users, can effectively mitigate the limitation. Given that a naming system is not just about a single global TLD, but also involves the ability to assign subnames where each name owner can have a registrar to manage these subnames, a standardized solution is necessary.

The proposed solution involves standardizing a function, such as `isValid`, that determines whether a name is valid according to its parent. This function would systematically check the validity of each name in a hierarchy. For example, consider the name `vault.0xYamen.eth`:

1. `vault.0xYamen.eth`: The function `isValid` checks if "vault" is valid under `0xYamen.eth` based on custom logic defined by the owner.
2. `0xYamen.eth`: The function then checks if `0xYamen.eth` is valid under the `.eth` registrar, reflecting whether the name has been renewed or not.
3. `.eth`: Finally, the function checks if `.eth` is valid at the root level.

For edge cases where names cannot be verified (e.g., the function is not implemented, the name is owned by an Externally Owned Account (EOA)), the system would assume a state of unknown and treat the name as valid.

By performing this validation before resolving a name, the system can significantly limit the persistence of outdated records, ensuring that each name in the hierarchy is appropriately verified according to its current status and the rules set by its parent. This standardized approach provides a robust solution to the record persistence issue, making name resolution more reliable and secure.

## 2. Addressing the Problem of Name Assignment

The challenge of fairly assigning names in decentralized systems like ENS has led to many ideas, including auction-based models and demand-based fees. Even Vitalik Buterin has discussed the need for more dynamic approaches in his writings.

An auction-based model for name assignment may not be the most relevant approach. As, even if an auction is initiated, the community witnessing it is often small, giving early participants an advantage. What is needed is a system that remains fair in the long run, which is where demand-based fees come into play.

In the Universal Name System (UNS), the registration period for names is limited to one year, allowing for regular reassessment of a name's value and relevance. UNS plans to implement a demand-based fees controller that will evaluate names based on their importance within the community and adjust fees accordingly. This system acknowledges that the relevance of names can change over time—some brands or entities may become less significant as years go by.

UNS emphasizes the importance of maintaining a flexible and adaptive system. Instead of offering indefinite ownership of names, UNS encourages periodic renewal, which aligns with evolving logic and community needs. This approach helps prevent valuable names from being locked under outdated conditions while still respecting the principles of decentralization and ownership. By requiring renewal, UNS fosters a more equitable distribution of names, ensuring that the system remains fair and responsive to change.

### **3. Addressing Voluntary Release of Ownership**

In the Universal Name System (UNS), the voluntary release of ownership is handled more effectively, especially with [.lyx](#) names. Unlike ENS, where putting the name ownership to market again is not possible, UNS offers a simple and secure process for owners to "burn" their names when they no longer need them.

This process includes two steps to ensure safety and prevent accidental loss. Once the owner starts the release, the name goes into a 90-day grace period. During this time, the name is not available on the market, giving the owner a chance to reclaim it if they change their mind or make a mistake. After 90 days, the name is fully released and available for others to register.

This feature gives users more control over their names while also keeping the system flexible, allowing names to be reused and potentially increasing revenue from new registrations.

### **4. Addressing UX Limitations**

The Universal Name System (UNS) has implemented several key improvements to address the UX limitations identified in ENS.

Firstly, in the UNS Registry, users can set resolver data directly from the registry itself. This enhancement eliminates the need for two separate transactions, which was required in ENS. As a result, users can register a name and set the resolver data in a single transaction, reducing both the time and transaction fees associated with the process.

Additionally, UNS has streamlined the process for transferring [.lyx](#) names, which are represented as NFTs. In UNS, whenever the ownership of the NFT changes, this change is automatically reflected in the registry. This means there is no need for a second transaction to update the registry, as was required in ENS. By directly linking the NFT ownership change with the registry update, UNS eliminates unnecessary costs and mitigates potential security risks that could arise from delays or discrepancies in ownership records.

# Conclusion

While this paper highlights several issues within the current ENS system, it is important to recognize that these challenges do not diminish the innovative nature of ENS as a pioneering system for hierarchical name assignment. ENS represents a significant advancement in decentralized naming services, and its contributions to the ecosystem are undeniable. The intent of this paper is not to criticize ENS but to identify areas where improvements can be made. By addressing these issues, we can pave the way for future naming systems to be even more robust, adaptable, and fair. The insights provided here aim to contribute to the ongoing evolution of decentralized naming, leading to structural enhancements that will benefit future implementations.