



INTRODUCTION

Ethereum domains is a distributed, open, and extensible naming system based on the Ethereum blockchain.

ENS's job is to map human-readable names like 'alice.eth' to machine-readable identifiers such as Ethereum addresses, other cryptocurrency addresses, content hashes, and metadata. ENS also supports 'reverse resolution', making it possible to associate metadata such as canonical names or interface descriptions with Ethereum addresses.

ENS has similar goals to DNS, the Internet's Domain Name Service, but has significantly different architecture due to the capabilities and constraints provided by the Ethereum blockchain. Like DNS, ENS operates on a system of dot-separated hierarchical names called domains, with the owner of a domain having full control over subdomains.

Top-level domains, like '.eth' and '.test', are owned by smart contracts called registrars, which specify rules governing the allocation of their subdomains. Anyone may, by following the rules imposed by these registrar contracts, obtain ownership of a domain for their own use. ENS also supports importing in DNS names already owned by the user for use on ENS.

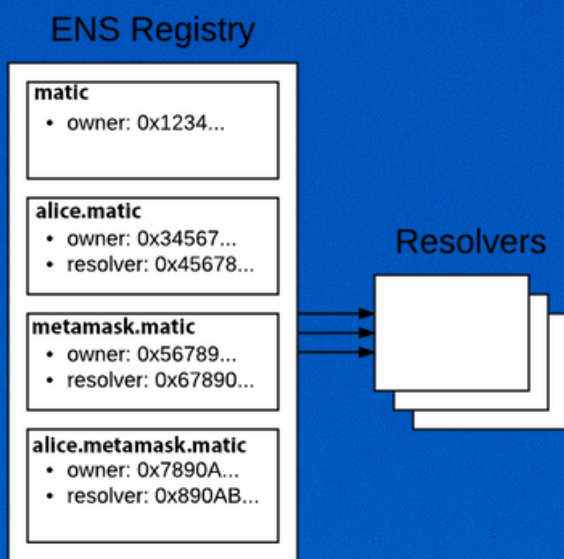
Because of the hierarchical nature of ENS, anyone who owns a domain at any level may configure subdomains - for themselves or others - as desired. For instance, if Alice owns 'alice.eth', she can create 'pay.alice.eth' and configure it as she wishes.

ENS is deployed on the Ethereum main network and on several test networks. If you use a library such as the ethers.js Javascript library, or an end-user application, it will automatically detect the network you are interacting with and use the ENS deployment on that network.

You can try ENS out for yourself now by using the ENS Manager App, or by using any of the many ENS enabled applications on our homepage.



ENS Architecture



The ENS registry consists of a single smart contract that maintains a list of all domains and subdomains, and stores three critical pieces of information about each:

- The owner of the domain
- The resolver for the domain
- The caching time-to-live for all records under the domain

The owner of a domain may be either an external account (a user) or a smart contract. A registrar is simply a smart contract that owns a domain and issues subdomains of that domain to users that follow some set of rules defined in the contract.

Owners of domains in the ENS registry may:

- Set the resolver and TTL for the domain
- Transfer ownership of the domain to another address
- Change the ownership of subdomains

The ENS registry is deliberately straightforward and exists only to map from a name to the resolver responsible for it.

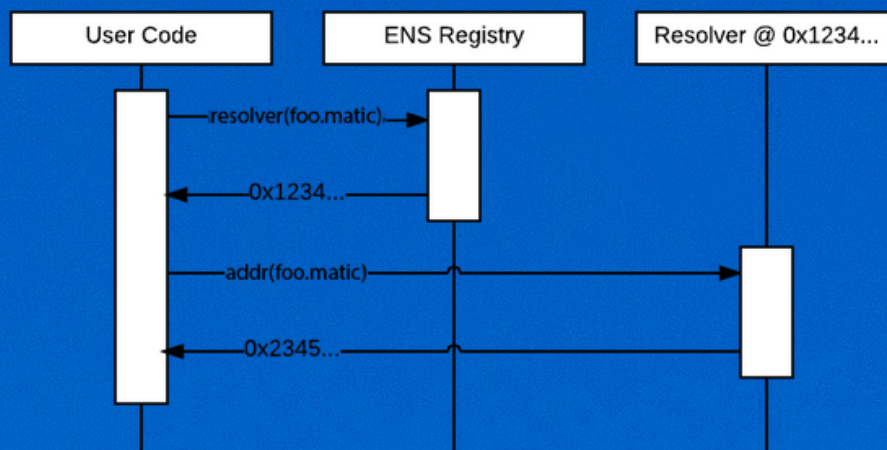
Resolvers are responsible for the actual process of translating names into addresses. Any contract that implements the relevant standards may act as a resolver in ENS. General-purpose resolver implementations are offered for users whose requirements are straightforward, such as serving an infrequently changed address for a name.



ENS Architecture

Each record type - cryptocurrency address, IPFS content hash, and so forth - defines a method or methods that a resolver must implement in order to provide records of that kind. New record types may be defined at any time via the EIP standardization process, with no need to make changes to the ENS registry or to existing resolvers in order to support them.

Resolving a name in ENS is a two-step process: First, ask the registry what resolver is responsible for the name, and second, ask that resolver for the answer to your query.



In the above example, we're trying to find the Ethereum address pointed to by 'foo.matic'. First, we ask the registry which resolver is responsible for 'foo.matic'. Then, we query that resolver for the address of 'foo.matic'.



NAMEHASH

Resource constraints in smart contracts make interacting directly with human-readable names inefficient, so ENS works purely with fixed length 256-bit cryptographic hashes. In order to derive the hash from a name while still preserving its hierarchal properties, a process called Namehash is used. For example, the namehash of 'alice.matic' is 0x787192fc5378cc32aa956ddfdedbf26b24e8d78e40109add0eea2c1a012c3dec; this is the representation of names that is used exclusively inside ENS.

Namehash is a recursive process that can generate a unique hash for any valid domain name. Starting with the namehash of any domain - for example, 'alice.matic' - it's possible to derive the namehash of any subdomain - for example 'iam.alice.matic' - without having to know or handle the original human-readable name. It is this property that makes it possible for ENS to provide a hierarchal system, without having to deal with human-readable text strings internally.

Before being hashed with namehash, names are first normalized, using a process called UTS-46 normalization. This ensures that upper- and lower-case names are treated equivalently, and that invalid characters are prohibited. Anything that hashes and resolves a name must first normalize it, to ensure that all users get a consistent view of ENS.