

Namespace Technical Workflow

Our stack consists of frontend dapp, backend application and smart contracts.

Smart contracts are used for listing and minting ENS subnames. An ENS name owner that wants to list on our platform will access the listing manager via the frontend part of the application and configure listing settings. The listing transaction is then executed by the NameMintingRegistrar smart contract, which will then store the listing configuration in NameRegistry. Additionally the listing may contain configuration that will determine whether a subname may be minted based on whitelist and token gated access, and whether a subname is reserved. This additional configuration will be stored in WhitelistRegistry, TokenGatedRegistry and ReservedRegistry.

Once the listings are stored in the above registries, the backend part of the application will listen to listing events and store the listing data, which will then be used to serve the subname search and minting features on our platform. Potential subname buyers will then be able to access our platform via the frontend and perform search queries for available ENS subnames. The backend will determine whether the queried subname is reserved, or whether minting is restricted based on the whitelist and token gated access. If the subname is allowed to be minted the backend will then return the listed price required to be paid by the subname buyer.

The minting transaction is then handled by the NameMintingController smart contract, which validates the transaction against the listing configuration in the above mentioned registries. If the validation passes, NameMintingController will deduct the payment amount from the buyer's wallet and will proceed with the call to NameWrapper to mint the subname for the buyer.

Additionally our platform also offers the ability to search and mint ENS top level domain names. The same search parameter that is used to query available subnames, will be used to perform the search for available ENS names. For the ENS name search our platform is utilizing the ensjs frontend library. If an available ENS name is found and selected for minting, the frontend dapp will call the ENS ETHRegistrarController. In addition to the ensjs library the frontend dapp is using the Thorin library to implement the application look and feel.

Another feature that the platform has implemented is SIWE, which can be used by subname owners to manage their account profile settings.

Smart Contracts Overview

- **NameWrapperDelegate**
 - makes calls to NameWrapper to mint subnames
 - prior to the listing, owner must give approval to NameWrapperDelegate to call setSubnodeRecord on NameWrapper
 -
- **NameMintingRegistrar**
 - handles name listing transactions and stores listing configurations in registries
 - supports single name listings and bulk listings
- **NameRegistry**
 - stores basic listing configuration which contains subname pricing information based on subname number of letters, mint deadline and payment recipient address
- **ReservedRegistry**
 - stores reserved names that are not available for minting, or can optionally be minted at special price
- **WhitelistRegistry**
 - stores whitelisted addresses that are allowed to mint for free or at a specified price
- **TokenGatedRegistry**
 - stores information on ERC20 and ERC1155 tokens that are required to be owned in order to mint subnames
- **NameMintingController**
 - executes minting transaction by calling NameWrapperDelegate which calls the NameWrapper setSubnodeRecord function
 - additionally delegates calls to upgradeable delegate contracts to perform minting subtasks:
 - **RegistryCalls** - retrieves listing data from the registries
 - **SubmintValidationCall** - validates the minting transaction against minting whitelist, reserved subnames, token gated access and minting deadline
 - **PaymentCall** - validates payment amount, deducts the payment and minting fees from the buyer's wallet, and transfer the earned amount to the payment recipient wallet specified in listing configuration
- **BulkWrapper**
 - in order to be able to list on the platform ENS names must wrapped, thus BulkWrapper will perform the wrapping operation by calling NameWrapper

- able to wrap single names and multiple name in a bulk transaction