**Explaining the Ethereum Namespace Auction**

Maurelian March 13 2017

*This post is to meant to inform the Ethereum community and solicit feedback. Questions and comments are very much appreciated and will be used to create a follow up FAQ. The first portion of the post provides the motivation, if you just want the nitty gritty, skip ahead to “***Ownership of names***”.*

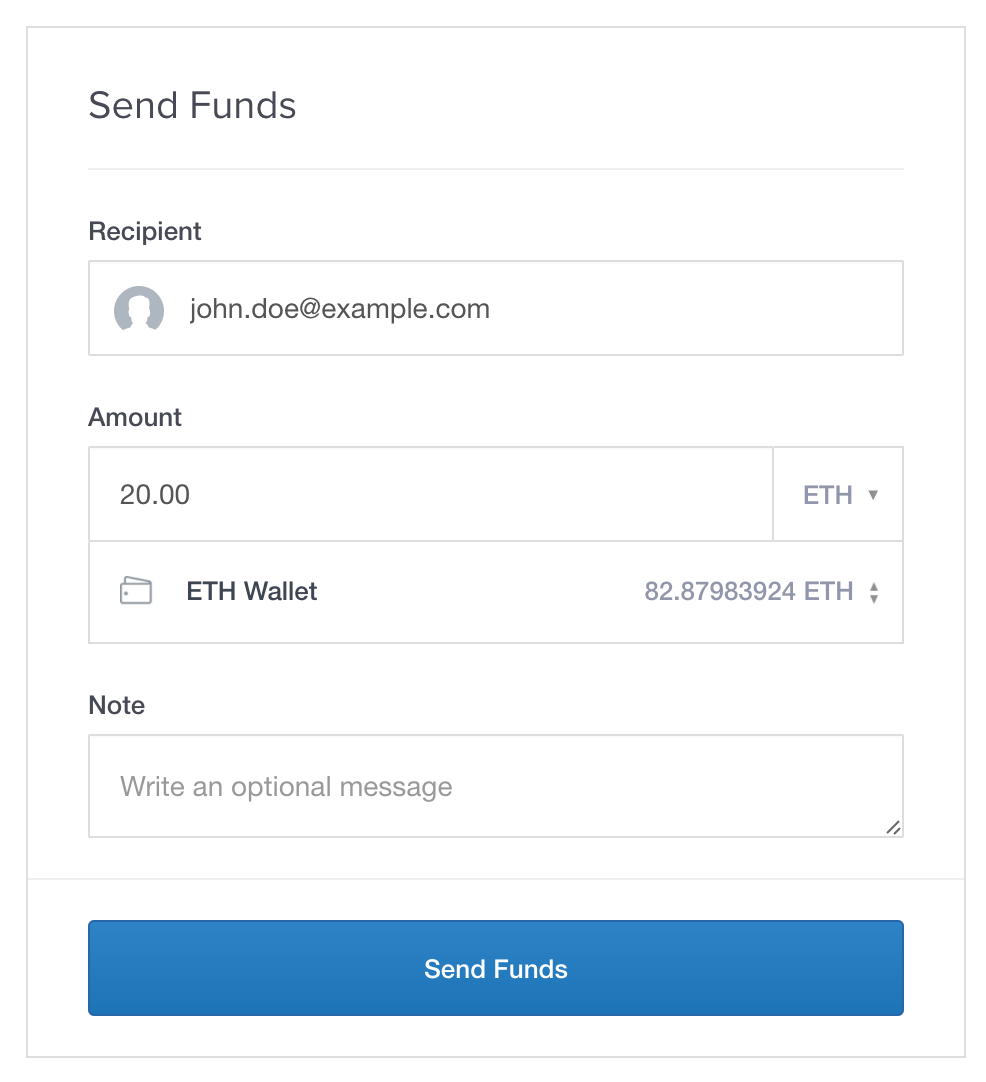
**Update: the**[**FAQ is here**](https://github.com/ethereum/ens/wiki/Frequently-Asked-Questions)**.**

**Ethereum Addresses are a barrier to adoption**



Welcome to Ethereum, here’s your name tag.

Anyone reading this is used to typical cryptocurrency addresses, and may even take them for granted. For normal humans however, the experience of receiving your *very own 160 bit hexadecimal encoded hash string* is not the ideal on-boarding process.



Easy.

Compare this to Coinbase, Circle, Paypal or Venmo; where you can send money to someone just using their email address or a username.

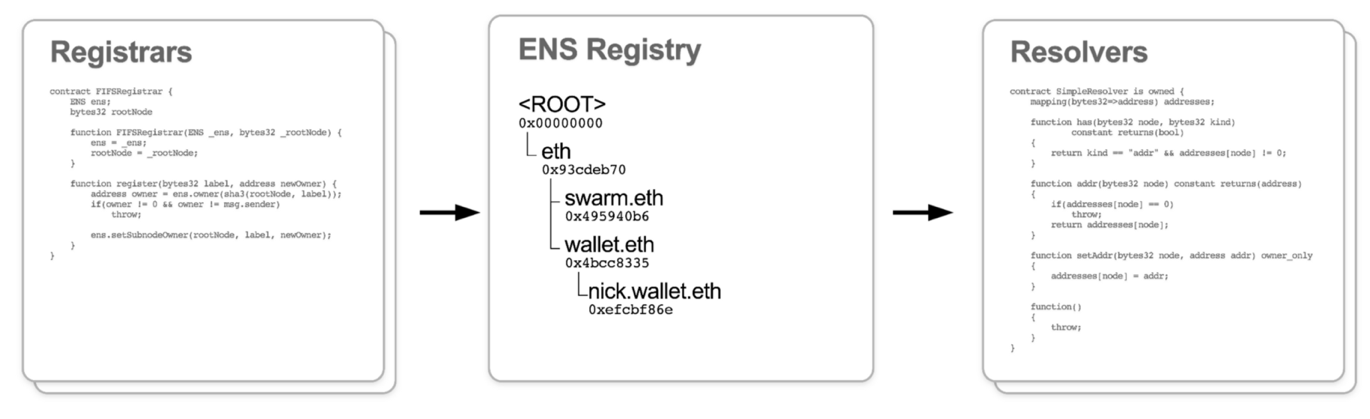
Now imagine sending ethereum transactions to people and contracts just as easily in Mist or any other Ethereum service, without relying on a centralized server to connect the name to the address.

**How the Ethereum Name Service Fixes Addresses**

The Ethereum Name Service (outlined in [EIP-137](https://github.com/ethereum/EIPs/issues/137)) brings human readable names to Ethereum. It will be the default name registry for the Mist wallet, and hopefully many other service providers.

Names in the Ethereum Name Service (ENS) look just like the familiar DNS addresses we use today, with “**.eth”** initially being the only top level domain (TLD) available. Just like you might browse to “www.example.com” you could send money to “wallet.example.eth”.

From a high level view, the ENS consists of 3 unique types of contracts:



From Nick Johnson’s [Devcon2 presentation](https://www.youtube.com/watch?v=pLDDbCZXvTE).

1. **The Registry contract** is the heart of the ENS, there is only one. It’s a simple contract; basically a big mapping of registered names to the address of an owner, and the address of a resolver contract.
2. **Resolver contracts** map a name to a “resource” such as an Ethereum address, or Swarm or IPFS hash. This is the contract Mist will get an address from when you say you want to send money to wallet.example.eth. Only the owner of a name can update the resolver’s records.
3. **Registrar contracts** own domain names in the ENS registry, and are responsible for allocating subdomain names within the namespace they control. The first ENS registrar will be the owner of the “.eth” top level domain name in the ENS.

**The challenges of registrar design**

The [registry contract](https://github.com/ethereum/ens/blob/master/HashRegistrarSimplified.sol), and [resolver contracts](https://github.com/ethereum/ens/blob/master/PublicResolver.sol) are elegantly in their simplicity. If you own a name, you can update their records for that name, otherwise they will simply ignore you.

Designing a registrar on the other hand, isn’t quite so straightforward. It involves subjective properties like user experience, mechanism design, fairness, and governance.

In a namespace, each name can only have one owner, making it a scarce, non-fungible property. Moreover, names are (usually) made up of words, which are powerful symbols: they can represent a brand, a movement, or your online identity across numerous platforms. It’s reasonable for people to feel a sense of entitlement to ownership over certain names.

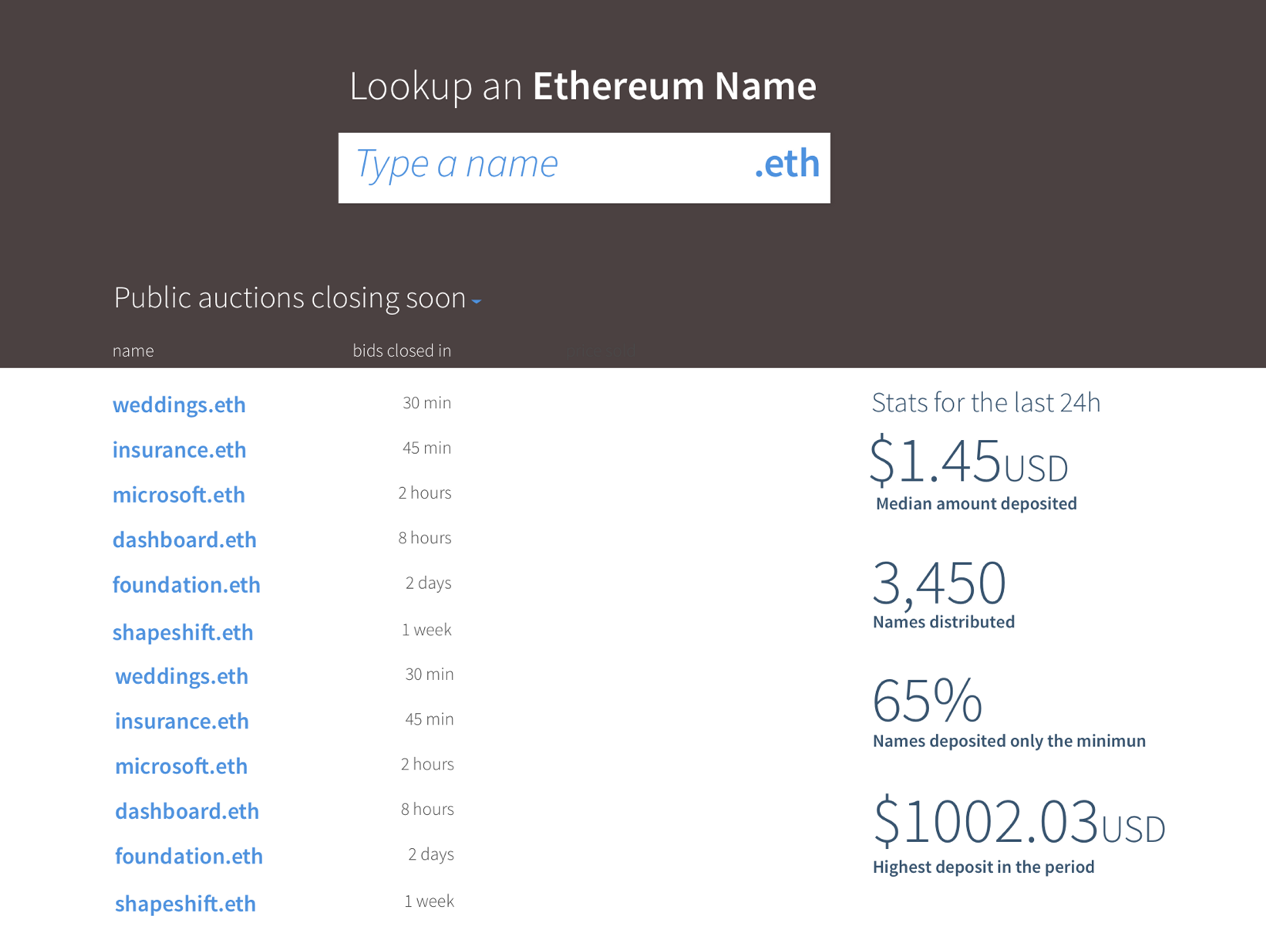
At the same time, end users looking up a common, well known name generally expect it to belong to the entity they associate it with. Who should control a name like **“microsoft.eth”**, **“bankofamerica.eth”**, or **“vitalik.eth”**? How should a registrar allocate these names? Should it be completely market driven, or should there be a process for dispute resolution?

After much discussion, the chosen strategy for allocating names is a compromise involving:

* an initially limited name space,
* a fully market driven approach,
* **an upgradeable Initial Registrar contract providing the ability to introduce a dispute resolution process in the future**.

The remainder of this article outlines the Initial Registrar’s functionality, and what you will need to know in order to register a name. I will point out the names of relevant functions in the Initial Registrar contract’s interface. For the full interface refer to [ERC-162](https://github.com/ethereum/EIPs/issues/162), and the full code in the [github repository](https://github.com/Arachnid/ens/blob/master/HashRegistrarSimplified.sol).

There will also be an app available to reduce the barriers to participation. You can view mockups of that [app here](https://projects.invisionapp.com/share/FE93G2K3Y#/screens/200024092).



**Ownership of names**

Names will be auctioned according to a process described below. The value paid by the winner will be *held in a deed* in exchange for ownership of the name. There is no recurring fee, or subscription required to own a name. After 1 year of registration, the owner of a name is free to release it, and the full value of the deed will be returned to them.

**Registration does not guarantee perpetual ownership**

The Initial Registrar will not include any dispute resolution process. However, some form of dispute resolution *may* be introduced in a permanent registrar which succeeds it. There are many options for how dispute resolution could be handled in the future, and new ones will certainly emerge after the launch of the ENS.

One possible approach, for example, would require domain holders to periodically adjust the value held in a deed to match the average increase in value of all domain names.

This creates the possibility that ownership of names may be revoked. Should that occur, **any funds held in the deed would be returned to the owner.**

It is also possible that dispute resolution will not be deemed beneficial, and changes introduced in the permanent registrar would minimal, limited to tweaks to the auction process.

**Transferring names**

The owner of a name may transfer ownership, along the ether held in the associated deed, to another account at any time. This will allow for secondary markets to form, providing the ideal path for a name to find an owner who can maximize the utility of a name.

**Service life timeline of the Initial Registrar**

The release of the Ethereum Name Service, including the Initial Registrar is planned for the first quarter 2017.

**Upgrading**

The Foundation will begin to accept proposals for a permanent registrar one year after launch. The Initial Registrar is expected to be replaced within 2 years of launch.

Owners of names in the Initial Registrar will be responsible for transferring their deeds to the permanent registrar. Options being considered to manage this transition include:

1. Requiring owners to transfer their ownership prior to a cutoff date in order to maintain ownership and/or continue name resolution services.
2. If the permanent registrar is lacking an entry, have it query the Initial Registrar to identify the owner

**Deadlines**

The following events and times will be hard coded into the contract:

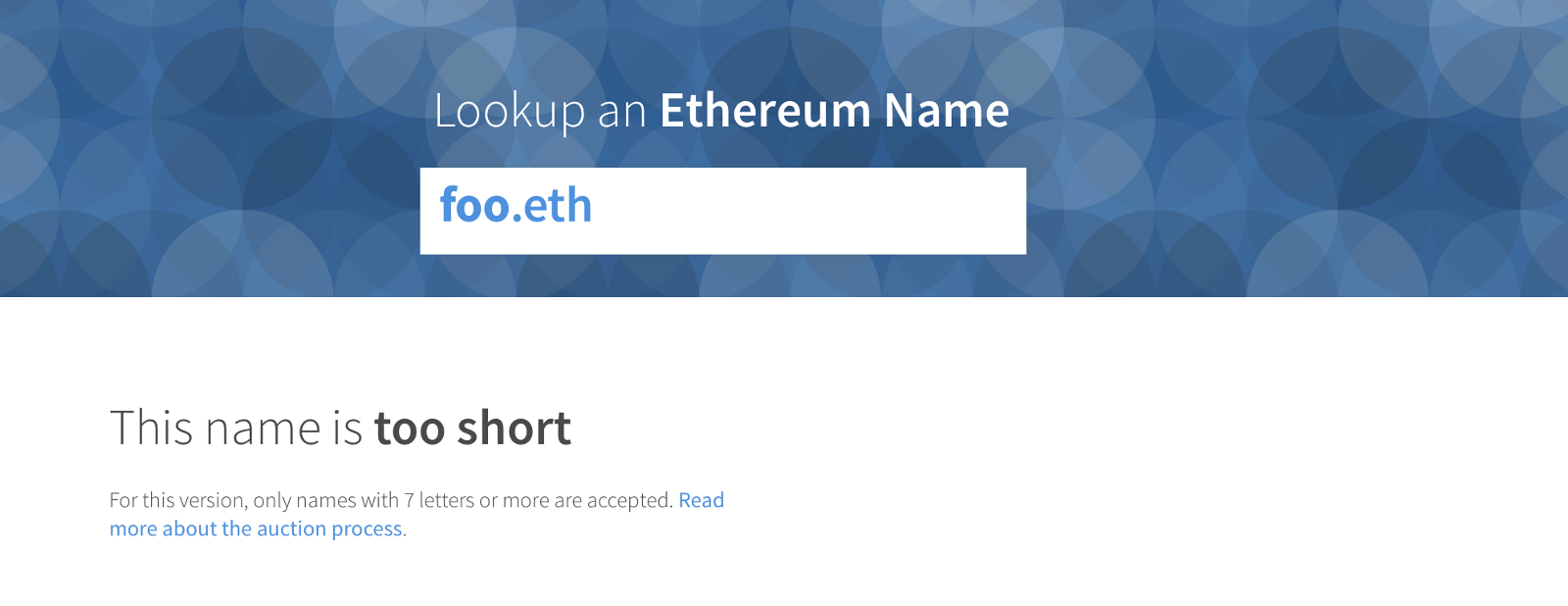
* **After 4 years, the Initial Registrar will refuse to register new names.**
* **After 8 years, funds held by the the Initial Registrar can no longer be released by owners, and will be irretrievable.**

Similar to Ethereum’s [difficulty bomb](http://ethereum.stackexchange.com/a/3819/4894), these hard dates force action and progress towards a permanent registrar.

**Names restrictions**

The only top level domain available will be **“.eth”**. Subdomains will be further **limited to 7 characters or longer**. For example:

* **“microsoft.eth”** will be **valid**
* **“google.eth”** will be **invalid**

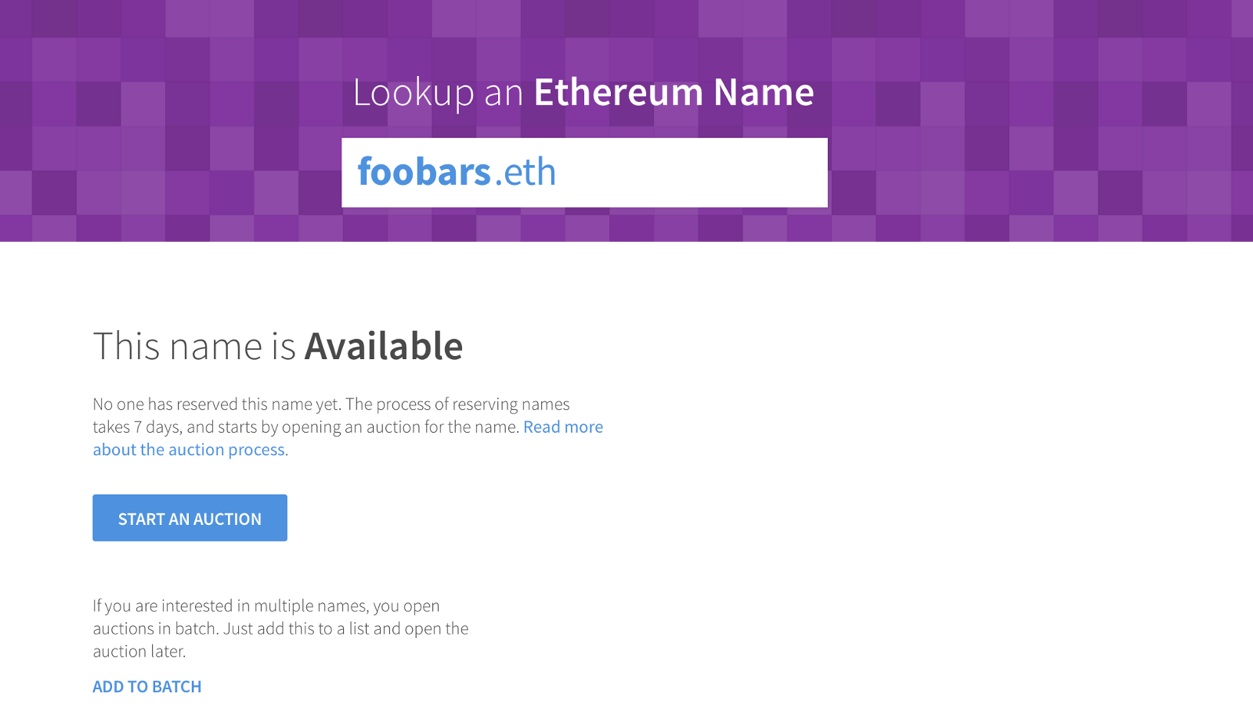


Note that subdomains of registered names have no restrictions. The owner of “**microsoft.eth”** can thus direct “**foo.microsoft.eth”**however they like.

This restriction reserves the shortest names (which are scarce, and often most valuable) for a permanent registrar with an improved approach to name allocation.

**The Initial Registrar auction process**

Anyone can start an auction for a name they are interested in by calling the registrar’s **startAuction()**function.

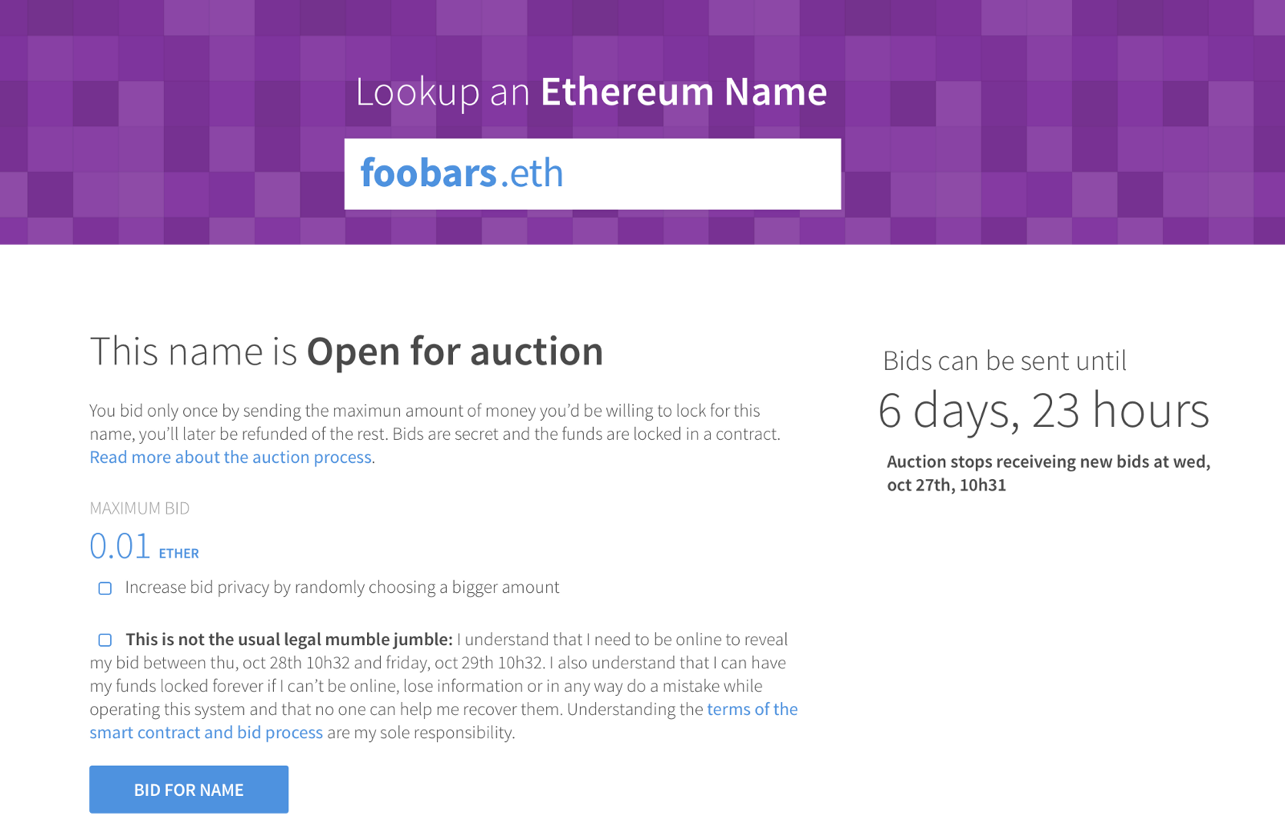


The Initial Registrar will auction names through a [Vickrey auction](https://en.wikipedia.org/wiki/Vickrey_auction); bids are submitted in a “sealed envelope”, the highest bidder wins the auction, but pays the price of the second highest bid. This format incentivizes bidders to bid what a name is worth to them.

The parameters of a bid are:

* a name,
* the value, in ETH, you are willing to pay for ownership of the name,
* the address which will own the name.

These parameters are salted and hashed (using **shaBid()**), and sent to the Registrar with a payment of ETH (using **newBid()**). Hashing the parameters makes it impossible to discern discern which name a bid is for.



**Minimum price**

The minimum price to register a given name is **0.01 ETH**. If no one else bids against you, that is how much you will pay.

**Auction durations**

During the normal operation of the registrar, the bidding period will last for 3 days. In order to allow people sufficient time to prepare for bidding, all auctions started during the first 25 days of the registrar’s life will extend until the end of the 28th day.

**The reveal period**

After the 3 days of bidding, is a 48 hour *reveal period,*during which bidders must submit the parameters of their bid (using **unsealBid()**). New bids are not accepted during the reveal period.

Bids revealed after the reveal period has ended will be invalid, and the value of the deed will be lost.

After the reveal period has passed the winner will be set as the owner of the name (using **finalizeAuction()**).

In the event that two parties bid exactly the same value, the first bid revealed will win.

**Costs of bidding**

Non-winning bidders will receive a 99.9% refund on the value bid. The remaining 0.1% will be burned. This creates a real cost to bidding on a large number of names, or highly valuable names.

**Learn more**

Have questions? **Please ask** by highlighting the relevant section, or commenting below.

**Other resources**

* [Nick Johnson’s Devcon2 Talk](https://www.youtube.com/watch?v=pLDDbCZXvTE) and [slides](http://dev/)
* [Auction app mockups](https://projects.invisionapp.com/share/FE93G2K3Y#/screens/200024092)
* [ENS implemented in LLL](https://github.com/zigguratt/ens-in-lll)
* ERC 137: [Ethereum Name Service Spec](https://github.com/ethereum/EIPs/issues/137)
* ERC 162: [Hash Registrar Spec](https://github.com/ethereum/EIPs/issues/162)

*Thank you to*[*Nick Johnson*](https://medium.com/@weka)*,*[*Alex Van de Sande*](https://medium.com/@avsa)*for proposing and writing the ENS’ contracts. Thank you also to Nick, Alex,*[*Jorge*](https://medium.com/@jmiehau)*and*[*Edu*](https://medium.com/@eduardo)*for reading and providing feedback.*