Ethereum Name Service
Hello!

Me: Brantly Millegan

Work: True Names LTD (Singaporean non-profit)

Open source project: Ethereum Name Service

Github: https://github.com/ensdomains

Website: https://ens.domains

Lead developer: Nick Johnson

Initial patron: Ethereum Foundation
Outline

● What is ENS
● How ENS Works
● DNS names on ENS
What is **ENS**
Ethereum Name Service

Complementary Internet naming protocol with the security, censorship-resistance, and programmability of the public Ethereum blockchain
Our **Goal**

Complement the DNS tech stack in parallel, expanding the capabilities of the existing DNS namespace, focusing on use cases not served by DNS (in some cases not possible on DNS)
Our Philosophy

Better infrastructure > competing namespace

- Support ICP-3 and ICANN jurisdiction over namespace
- ENS-native TLD .ETH
  - Experimentation with full benefits (ICP-3.5)
  - Mostly for blockchain projects
- DNS namespace integration
  - .XYZ, .LUXE, .KRED, .ART, .CLUB integration
  - Soon all DNS TLDs
How ENS Works
How ENS Works

No servers

Runs entirely as a set of smart contracts on the Ethereum blockchain

No permission needed to use the system, just need access to Ethereum blockchain
How ENS Works

**Registry**: SC that holds names, owners, resolvers, TTL

**Resolvers**: SC that stores and serves records

**Registrars**: SC that creates and distributes names based on certain criteria (e.g. payment)

To make any change (register name, set records, transfer ownership, et al) the user submits a certain kind of Ethereum tx or set of txs
How ENS Works

Resolutions/lookups (of records, whether name is available, et al) do not cost anything because you’re simply looking up information on the public Ethereum blockchain.

Can be done locally (Ethereum node on local machine), with an Ethereum gateway service (Cloudflare, Infura, et al), or with a light clien (still in development).
ENS Records

“Public Resolver”
- **Address:** Ethereum address
- **Other Addresses:** Other cryptocurrencies
- **Content:** IPFS, Swarm, Tor .onion
- **Text records:** Arbitrary records
- **DNS records** (not available in Manager UI)

You can also make your own custom resolver
ENS Records

If you want to see example of records for yourself:

app.ens.domains

Search for “brantly.xyz”
ENS vs. DNS

- Simpler infrastructure (servers → smart contracts)
- Zero down time, can’t DDOS attack
- Decentralized, censorship-resistant
- Built-in cryptography and individual ownership
- Better security
- Programmability, ownership by smart contracts
- Interactivity with other Ethereum smart-contracts
ENS vs. DNS

Security and censorship-resistance
- Cryptocurrency addresses
- IPFS websites

Programmability/interactivity on Ethereum
- Names owned by smart-contracts, or groups (e.g. DAO, multisigs, et al)
- Used as non-fungible items in (e.g. trustless loan)
DNS names on ENS
DNS names on ENS

Expands usefulness of existing namespace

No cost
Two Methods

Claiming TLD

Claiming 2LD
Claiming **TLD**

.LUXE, .KRED, .ART, .CLUB

Submit proof, ENS root keyholders assign ownership

Original plan with nic.TLD

Can make revocable or irrevocable
Claiming 2LD

Currently works with .XYZ names (e.g. brantly.xyz), soon all DNSSEC-enabled names. (Side benefit is it incentives use of DNSSEC)

- Create _ens.example.TLD
- TXT record: 
  a =0x983110309620D911731Ac0932219af06091b6744
- Submit DNSSEC proof to smart contract (NSEC to delete)
Public Suffix List

Will start with just TLDs.

But lots of 2LDs (and +3LDs) act as TLDs.
Public **Suffix List**

Two possible methods of dealing with this:

1) Maintain a list of suffixes in smart-contract, manually update as needed, using publicsuffix.org as reference

Lots of exceptions, lots of manual work, may not always be perfectly up-to-date
Public **Suffix List**

2) Always allow a domain to be claimed as long as higher level isn’t taken.

Can we count on no registrars (particularly ccTLDs) allowing someone to register \_ens.TLD or \_ens.public.suffix?
Useful Links ...Questions?

- **Website**: ens.domains
- **Manager**: app.ens.domains
- **Docs**: docs.ens.domains
- **Medium**: medium.com/the-ethereum-name-service
- **Twitter**: @ensdomains
- **Email**: brantly@ens.domains