MVP of ENS on L2 with Optimism: Demo Video + How to Try It Yourself

Nick Johnson

Follow

--

Listen

Share

At the ENS workshop earlier this week, I demoed the first MVP of ENS's L2 support, showing how an ENS name can be resolved on Optimism.

This end-to-end demo is the first instance not just of resolving an ENS name using a layer 2 solution but also the first demonstration of an entirely new system for trustless communication between applications on layer 1 and different layer 2 solutions.

This system, based on an idea by Vitalik Buterin and described in detail inthis medium post, defines a common way of interacting with layer 2 platforms such as Optimism, so that applications can query them without having to know the specific details of each platform's implementation — and without any additional trust assumptions.

This makes it possible to store ENS records on layer 2 platforms, permitting easy and cheap updates to ENS data without paying Ethereum gas fees. With a simple update to the process of resolving an ENS name, all ENS clients will be able to resolve these names — without having to support each L2 platform individually. In a rapidly evolving ecosystem with many L2 solutions reaching maturity, this flexibility is essential in allowing users of ENS to choose the scaling solution that works for them.

Applications beyond ENS include offering a generic interface for wallets to display the balances of tokens stored against the user's account on L2 platforms, and even to transfer those tokens back to L1, again without the wallet having to individually support each L2 platform.

How it works

The core of this new functionality is an "L2 gateway" — a server that speaks a common protocol to answer questions about data stored on a layer 2 platform. This is combined with a smart contract on Ethereum in a three-step process:

1. The DApp calls the Ethereum contract with the query they want to make. The contract responds with the URL of the L2 gateway, and a prefix that the data returned by the gateway must have.
2. The DApp then calls the L2 Gateway service using the URL returned in step 1, sending the same query data they sent the contract in step 1. The gateway looks up the information on the L2 platform it interfaces with and returns it to the caller. The caller checks that this information starts with the prefix it was given in step 1.
3. Finally, the DApp calls the Ethereum contract again, passing in the data the gateway returns. The contract interprets the gateway's data and validates it, returning the requested information to the DApp.

Step 3 allows the Ethereum contract to validate the data returned by the L2 Gateway, ensuring that it cannot return inaccurate results. For example, with Optimism, the data returned by the gateway can include cryptographic proofs that the query response is part of the Optimism state, and the Ethereum contract can validate this by querying the Optimism contracts on Ethereum.

This model also facilitates securely fetching information from other external systems not commonly considered to be Layer 2 solutions — for example, a gateway that fetches information from a merkle tree, whose root is stored in the Ethereum contract that validates the responses.

Try it yourself

The source for the ENS workshop demo is available on GitHub [here](https://github.com/ensdomains/l2gateway-demo).

It demonstrates an end-to-end implementation of the above protocol for resolving ENS names using the L2 Gateway solution described above and Optimism's integration repository, which deploys a local instance of Optimism's layer 2 platform on your computer.

The included webapp demonstrates the new ENS resolution process in step-by-step fashion, so you can see what is happening at each step.

Next steps

Going from an MVP to a fully fleshed out and mature standard is a large task which we're fully focused on at ENS.

Our first step will be to write up a standard for the gateway interface, so that ENS and other projects can build on top of it. After that, we can write up a specification for how to use this new protocol with ENS, and implement it in the ENS client libraries used by wallets and other applications that resolve ENS addresses.

Interested in participating in the development of the L2 gateway standard, or of ENS's next evolution? Join us on our Discourse forum, or our Discord!

Follow us

- Subscribe to our email list
- Follow us on Twitter
- Get support help on Discord
- Join the discussion on our forum
- Learn more on our website