

L2 Funding Proposal

Seeking funds for a next generation L2 explorer

Blockscout is a primary open-source explorer for the Ethereum ecosystem. Over 100 chains currently use Blockscout in one capacity or another. This includes testnets that can quickly spin up an instance (Kiln network is an excellent example), newer and mature ecosystems looking for a decentralized alternative to Etherscan, and more recently L2s in need of a customizable explorer for batching attributes, L1 <-> L2 transaction transparency, and other data unique to the rollup environment. Current L2s using Blockscout include the Optimism [Bedrock Beta](#) & [Goerli](#) testnets, [Arbitrum Nova](#), [Metis](#), and [Boba](#).

While these projects have been able to get Blockscout up-and-running quickly, and in a basic capacity, there is a lack of advanced features available for L2 projects. This has led several L2s to start with the Blockscout UI to setup basic configs, then move operations to Etherscan (while still potentially using some features of the Blockscout API). Users and projects expect Etherscan-like capabilities that Blockscout does not yet support.

However, what Blockscout does support is open-source code and ecosystem transparency. In order to more quickly reach parity with closed-source explorers and offer users a superior experience for L2 exploration and ultimately a more cohesive and blended L1/L2 experience, Blockscout can greatly benefit from additional grant funding.

What is Needed

L2 Data Views

Blockscout's base configuration is geared towards L1 EVM implementation, centered around block exploration. To incorporate L2s effectively, batches need to take precedence. In particular batch attributes such as Index, L1 Timestamp, Size, L1 tx hash, L1 block number, and Batch root should be easily explorable.

L1 to L2 transactions need to be clearly defined along with access to the current fraud proof window when appropriate. Transaction batch and state batch details should be included in an easy-to-explore format. Transaction fraud proof windows should be displayed along with countdowns, and if already executed, the transaction should be linked to its L1 counterpart.

Contract verification should be easy and available to rollup users, and system contracts (including the rollup contracts themselves, bridge contracts and token contracts) should all be verifiable.

These are simply some starting points for L2 data incorporation. As we continue our research into L2 systems we will further define the required data and views to best display digestible information for both the end user and projects requiring access to available RPC methods and cross-L2 exploration tools. General specs will be developed along with customization options for different L2 solutions.

?

Blockscout UI features and enhancements

Blockscout is currently planning a major UI upgrade. New features will include a My Account section with custom tagging and custom ABIs, an apps integration, and an interface overhaul. Users are requesting a more modern interface with more features in-line with the Etherscan experience. This upgrade should help Blockscout attract new users and keep current users by providing a vastly improved explorer experience.

The work is ongoing and can be greatly expedited with additional resources. We feel this will remove a major blocker to more adoption of an open-source explorer. You can see some of the improvements on our [Goerli testnet implementation](#) by logging in and creating an account.

?

?

?

Developer experience improvements

Once the L2 framework is established, we will work to provide excellent code documentation, usage examples, and improvements to the deployment process. Customization of basic instances can be difficult for projects without a dedicated team, and we will work to streamline this process and make it more intuitive. In addition, we are introducing new microservices written in Rust with plans to continue transitioning to this more developer-friendly language from the current Elixir implementation.

We are also incorporating analytics into the interface to include additional transparency for L2 transactions and smart contract interaction.

Plans/Timeline

This is a basic outline of our proposed process, open for discussion and subject to change.

Month Tasks
1 Resource planning, hire additional developer(s). Research basic requirements, specs, definitions, start on a framework for general L2 data integration.
2 Custom fields and views integration, testnet instance deployments, data indexing.
3 Begin UI enhancements, focus on batch data views. Data optimization, RPC extension Bring in front end dev.
4 Extend enhancements to other views, incorporate account functionality, focus on contract verification tools and functionality.
5 L2-specific UI improvements, QA, analytics integrations.
6 Complete new generalized UI for L2 explorers, documentation, rollout and customization features. Bring in documentation help.

Why Blockscout?

As an open-source project with 5+ years experience in the space, Blockscout is uniquely positioned to extend the current explorer implementation to serve the L2 ecosystem. We are already working with various L2s including Optimism, zkEVM, and zkSync to gather requirements on their implementations and incorporate these learnings into a next generation L2 explorer. Additional grant funding will accelerate and improve this process immensely.

Last updated 7 months ago