Hi guys, Zhivko from LimeChain here.

This proposal was first discussed in the Research Forum.

## Abstract

This governance proposal suggests the development of an Interactive Reference Specification website for rollups, with Arbitrum being the first one. This website would serve as a valuable resource for developers and provide clarity on the differences between Arbitrum and Ethereum L1 EVM, as well as between different rollups. The website would be extending upon the idea of evm.codes as a base but would incorporate information on rollups (differences from L1, gas costs, custom precompiles, native precompiles support, system contracts, properties such as costs, latency and finality of the native L1 ↔ L2 messaging protocol provided by the given rollup etc).

## Motivation

This proposal comes from LimeChain - a blockchain development company building apps and infrastructure since 2017.

As blockchain devs, we sometimes find it challenging to assess the differences between developing on L1 versus a given rollup, or between different rollups. Additionally, it is unclear what custom precompiles exist on Arbitrum, which precompiles are supported, what L1 state is exposed on a rollup, what system-level contracts exist, what the gas costs, latency and interface are for L1 - L2 messaging and how EVM gas pricing has changed etc.

Although that information is usually present in the rollup’s documentation, it is often not easy to find and surface.

As many of you, we believe that the future of Ethereum is rollup-centric, however, developer documentation and resources around rollups are yet to be improved.

## Rationale

With this proposal, we want to make it easier for 1) L3 developers using Arbitrum Orbit 2) dApp developers building on Arbitrum and 3) auditors to leverage the Arbitrum stack.

## Specification

The website will start as a fork ofevm.codes (NextJS + React). The team will extend the UI with the following components

### General Overview

An overview of L1 and the supported rollups (starting out with Arbitrum). For each supported rollup, there would be a succinct description providing information about:

- The type of the rollup, the latency of data availability (how much time it takes for the sequencer(s) to post L2 data on L1) and the finality
- Does it have a modified EVM and if yes, a brief description of the modified components
- Does it have modified RPC endpoints and if yes, a brief description of them
- Does it support a native L1 ↔ L2

### Opcodes

The view will focus on any changes to the EVM opcodes of the Rollup, compared to the original L1 EVM. Changes include opcode not being supported or its gas being changed.

### Precompiles

- The view will differentiate between “native” to Ethereum L1 precompiles and “system” precompiles introduced by the Rollup. Information on whether all expected precompiles (native) are supported and whether there are changes to the gas schedules (many of the zkEVMs either do not support precompiles or have their gas costs modified).
- A list of all additional “system” precompiles and contracts will be documented. Many rollups (including Arbitrum) introduce peripheral system contracts such as the ArbOS Precompiles

### Messaging

Rollups usually introduce native L1 ↔ L2 messaging. In this section, we will outline the properties of the messaging, such as latency for delivery and gas costs (for both L1 → L2 and L2 → L1 directions).

## Steps to Implement

LimeChain will leverage its in-house team to build the website. The current specs require 2 full-time front-end developers, 1 UX/UI designer and a blockchain developer to research and outline the content for the website.

The website is yet to be named/branded. We’re currently leaning towards “rollup.codes”.

### Timeline

The development of this website (incl. the content for Arbitrum) will take 4-6 weeks with the aforementioned team. Starting date would be set once the community sentiment and scope are clear.

### Future Development

The website will be extended to provide information not only on Arbitrum but the rest of the rollups in the ecosystem.

The team will prioritise support for new rollups based on community interest and activity.

### Compatibility Scanner/Checker

Users will be able to provide their smart contracts to a scanner that will provide information on the compatibility of the code with various supported rollups.

### Overall Cost

The requested funding for this proposal is 15,000 ARB tokens. This funding will partially cover the development cost. However, what’s more important for us is the support from the Arbitrum DAO and the community.

Feedback on the scope and features is deeply appreciated!