

It's my second time posting here, so I apologize for any violations of decorum and welcome corrections.

tldr

If we fragment the state into hashes per-user (or per discrete piece of globally shared state with its own update rules, which might include things like LP or debt positions), we can achieve faster finality than monolithic rollups

. Although the cost reduction is not as much as a monolithic rollup, fast finality is highly advantageous for DeFi applications, which is my main [area of interest](#).

Detail

Conventional optimistic rollups include a large amount of data in a single state root. The constraints around rollup finality as far as I understand are as follows:

- The challenge period must be AT LEAST as long as the time to finality of L1 in order to avoid censorship/reorg attack/multi-block MEV.
- The challenge itself might take AT MOST as long in blocks as Nth root of the L2 state, where N is the width of the state tree.

The shortest time to finality therefore requires that the Nth root of the L2 state be verifiable in a single L1 block at reasonable cost even during periods of congestion. This is easy to do if each user has their own state root, which includes their balances of any tokens or deposits, while each contract like an LP position would store only the state that is shared across all users and also have its own state root.

A shared bridge contract/[based sequencer](#) allows for a set of optimistic roll-wallet users to use any desired contract logic and swap any number of tokens with each other for the flat mainnet gas cost per-user of updating their state roots.

With a withdraw delay/challenge period equal to the time to mainnet finality + a cushion equal to the expected number of blocks that a proposer could monopolize, the roll-wallet or other roll-app can still finalize much faster than a monolithic rollup and so offers superior interoperability with external liquidity.

Conclusion

This idea represents the opposite end of a spectrum from monolithic rollups. In the middle there is also some [interesting space](#). I greatly appreciate those who share related ideas and resources.