Ethereum adopted a rollup-centric roadmap a few years ago. Since then, the community has come up with a variety of different rollup designs, all aiming to scale our beloved blockchain in one way or another.

Today, we want to elaborate on a new rollup design that we introduced a few months ago. We believe that this new design comes very close to being a native Ethereum scaling solution. We call it a Based Booster Rollup (BBR).

Let's take a look at the Based Booster Rollup vision.

Based and boosted

Based rollups are arguably the most effective, decentralized, and Ethereum-aligned ( ) way of running a rollup. By delegating the sequencing job to L1, a based rollup inherits the L1 decentralization, simplicity, and liveness, among other benefits.

In other words, based rollups are what Ethereum needs to scale natively instead of introducing new risks with centralized or semi-centralized sequencing.

But what if you could amplify the benefits of a based rollup by adding out-of-the-box, native Ethereum L1 dapp scaling to it? That's exactly what booster rollups offer.

Imagine deploying your dapp once and having it scaled automatically across all L2s. If extra blockspace is needed, more booster rollups can be added without any additional setup work. Put another way, no significant extra work for devs, no redeployment costs, no additional worries.

In simple terms, booster rollups work similarly to adding extra CPUs/SSDs to your laptop: They make your computer more powerful, allowing apps to work faster and grow bigger.

For the more technical folk, here's a one-sentence technical summary of what booster rollups do: Booster rollups shard the execution of transactions and the storage.

If you're interested in how booster rollups work technically, read our Co-founder and CTO Brecht Devos' posts on the topic on ethresear.ch:

- Cross layer communication: Trivially provable and efficient read access to the parent chain
- Booster rollups - scaling L1 directly.
- Booster rollups part 2: ZK-EVM as a ZK coprocessor.

Cross layer communication: Trivially provable and efficient read access to the parent chain

Booster rollups - scaling L1 directly.

Booster rollups part 2: ZK-EVM as a ZK coprocessor.

Here's also an X thread explaining the thought process behind booster rollups.

Now let's see why based rollups and booster rollups are a perfect fit for each other.

Based Booster Rollup (BBR) vision

Booster rollups are indiscriminate. Any rollup, whether optimistic or ZK, with the necessary booster functionality can become boosted. However, not all rollups need to be fully boosted to still allow L2-only functionality where it makes sense.

Boosting a based rollup makes the most sense if you want to achieve native Ethereum scaling. By allowing L1 validators to propose blocks for the whole boosted network, you're scaling Ethereum out of the box. This means that a Boosted Based Rollup retains the advantages of a based rollup and

adds direct Ethereum scaling.

BBRs also have the potential to solve the fragmentation problem that all current rollups face (and create) — again, without

losing L1 sequencing and all the benefits that come with it. By adding atomic cross-rollup transactions between all L2s in the booster network, BBRs can offer everything that's needed to scale Ethereum the way it was imagined originally — in one place.

We think this is a very powerful design and one that comes very close to scaling Ethereum natively. That's the Based Booster Rollup vision.

What does it all mean?

Let’s recap and see what advantages a BBR brings to the table.

For users, the BBR approach means that they no longer need to worry about fragmentation and jumping from one L2 to another: All of their favorite dapps will be across all L2s.

Of course, the BBR design would significantly reduce transaction costs and increase throughput. Users would get to enjoy what they deserve — a scaled and secure Ethereum.

For devs, the BBR design allows them to boost their dapps without the need to redeploy to all L2s. Instead, just deploy your dapp once on L1, and you’re done — your dapp is automatically scaled across all boosted L2s deployed now and in the future.

.css-ovd0pw{position:relative;height:100%;width:100%;}

.css-istob7{height:100%;width:100%;position:absolute;max-width:100%;max-height:100%;}

.css-87l67c{padding-bottom:55.55555555555556%;}

.css-ovd0pw{position:relative;height:100%;width:100%;}

.css-istob7{height:100%;width:100%;position:absolute;max-width:100%;max-height:100%;}

.css-87l67c{padding-bottom:55.55555555555556%;}

.css-istob7{height:100%;width:100%;position:absolute;max-width:100%;max-height:100%;}

.css-87l67c{padding-bottom:55.55555555555556%;}

Wen?

When can we expect to see the first BBR go brrrr?

Right now, Taiko is building towards introducing contestation into the protocol. In our next testnet, we’ll test how a Based Contestable Rollup (BCR) mechanism, or at least parts of it, could work in practice. The BCR design is the design that we’re going to implement on our upcoming mainnet. We’ll have more information on that later.

After launching BCR on mainnet, which is our first major milestone, we’re planning to either upgrade our protocol to BBR or launch a separate BBR L2 as the second major milestone.

We believe BBRs have the potential to take Ethereum scaling to a whole new level. If there are teams who are interested in the BBR design and would like to work together on technical specifications and implementation, please reach out.

Also, take a look at the first Taiko Improvement Proposal (TIP):TIP-0001: Support reading and writing to the parent chain, which is a prerequisite for booster rollups, and come discuss with the community.

Stay tuned for more details in the coming months!

Join us

Explore open positions on our job board.

Follow us

Get the latest from Taiko:

- Website: https://taiko.xyz.
- Discord: https://discord.gg/taikoxyz.
- GitHub: https://github.com/taikoxyz.
- Twitter: https://twitter.com/taikoxyz.
- Community forum: https://community.taiko.xyz.
- Youtube: https://www.youtube.com/@taikoxyz.

Website: https://taiko.xyz.

Discord: https://discord.gg/taikoxyz.

GitHub: https://github.com/taikoxyz.

Twitter: https://twitter.com/taikoxyz.

Community forum: https://community.taiko.xyz.

Youtube: https://www.youtube.com/@taikoxyz.

Contribute

Contribute to Taiko on GitHub and earn a GitPOAP! You will also be featured as a contributor on our README. Get started with the contributing manual.