Mikko and Alex kicked off the Community Call #6

.

Team updates

Updates from Mikko

- Many Taiko folks are currently in Warsaw and Korea at the events. ETHSinagpore and Token 2049 are coming very soon as well. Even more Taiko folks will be there. All events can be found at taiko.xyz/events.
- On Taiko L2, around 5,500 proposers, 831 provers, and over 10M transactions. On Taiko L3, 4,010 proposers, 512 provers, and 2.8M transactions.

Updates from Alex

- Lisa organized the Taiko Research Day, an event where engineers talked about Rollups and zk-related stuff, such as Halo2, SGX, cross-chain dapps, etc. Check out the recordings on the Taiko YouTube channel.
- RJ published more interviews with Taiko people – this is a way to learn more about the team. Check the last episodes with Dave, Head of DevRel, and Mamy, ZK engineer.

Updates from Daniel

- Within the last month, we worked on optimizing our previous code base. We redesigned some features. For example, the prover is now selected off-chain. That is, proposers can negotiate the deal directly with potential provers.
- Internally, we're currently running the next devnet. The goal is to launch the next testnet, Jolnir L2, later this month (September). Two live testnets, Eldfell L3 and Grímsvötn L2, will be deprecated when the new one is launched. We will start with the same zk prover as we have now, but the zk coverage will go up after the launch. We are currently testing a new prover and still need more time. We expect this protocol version to be very close to its final version. However, there will be an update for EIP-4844 when it's live.
- For those who want to participate as provers and offer their services to proposers, you need to learn the API endpoint specs from our codebase and then publish your server endpoint on our website. We will have a page for it, and provers can submit a PR to be added.

Patryk intro

- "I am Patryk. I live in Warsaw, Poland. I joined Taiko two weeks ago as a software developer oriented on Intel SGX technology that we are considering using at Taiko as an extra layer of security next to the zk proofs running Ethereum client inside the SGX enclave. Before that, I used to work as a backend engineer, most recently in a startup creating an SGX-based privacy-preserving crypto lending platform."

Updates from Brecht

- It's a good moment to introduce multi-prover: it is hard to say that a single prover is completely bug-free and there are no issues, especially with zk proofs, as they are hard to build and it's easy to forget some constraints that will allow to create a proof for an invalid state transition. Of course, we are doing our best not to make it happen. But the complexity is so high that it is impossible to prove that there are no issues.
- The plan is to add several more types of proofs that will prove the block is valid with the correct post-state. We currently have zk proofs. But if we add proving systems that work differently, if there is a bug in one of them, it's unlikely that the same bug is exploitable in other proving systems. So, one more type of proof we are going to use is SGX. And in the future, we will probably add multiple zero-knowledge proofs (e.g., based on SNARK and STARK). Fraud proofs can also be used, but they are less interesting for us as they have a longer time to finality (up to a week).
- We added all the missing stuff in the EVM circuit within the last month. But some extra testing is still necessary. We are also working on optimizations for Halo2 and finishing up some circuits, such as block hash and RLP circuits. Some small improvements were made to the Merkle Patricia Tree and the "circuit tools" (how we write and review circuits). Krzysztof is doing some fuzzing on the bytecode circuit and some other circuits. Patryk is working on running geth inside the SGX, and after that, he'll also run reth (rust client implementation) inside the SGX.

Updates from Einar

- Einar said that was his first Community Call. He is currently working on some prover bottleneck optimization, such as MSM. If someone looked into the guts of the prover, there are three main bottlenecks: MSM (Multi-Scalar Multiplication), FFT (Fast Fourier Transformation), and NTT (Number Theoretical Transform). There are many techniques to optimize them, and many collaborations happen in the wide zk community: Taiko learns from the best, and they learn from Taiko. Einar added he was also working on switching to a different kind of coordinate system that

makes the representations of elliptic curves Taiko uses more efficient.

Updates from AJ on the grants program

- AJ said he was coordinating the grant program and wanted to give a quick update on it: Yesterday was the last day of submission, and we got many applications at the last minute. The number of submissions is close to one hundred. For the current grant cycle, applications are closed. We're already reviewing and progressing to the second stage with some of the applications. In our blog post, we promised to be back by the middle of September. We hope we can do it. But if the timing changes a bit, we will keep you posted.

Questions

- Are any improvements or new features planned for simple Taiko node (e.g., implementation of gas tip cap)?

Dave (who usually takes this type of question) wasn't at the call because of Korea events. Daniel mentioned that the team is currently working on supporting not only Docker Compose but also Kubernetes. But Daniel didn't have any info on delivery time.

- Will the target delay feature remain in Alpha-5?

The target delay feature is related to the previous proving system and will go away after the prover fee system is updated.

- How will rollups use composability?

As for now, there are different delays between different rollups. For ZK-Rollup, one has to wait until the proof is in its place to bridge between two rollups trustlessly. For optimistic rollups, you have to wait even longer (up to a week). One of the ways to do cross-chain communication seamlessly was described by Brecht in the "Multi-layer dapps" video.

- Alpha-3 used proof racing, and Alpha-4 used a staking-based approach. Which of these designs will be used after mainnets and why?

Taiko's goal is to have cheap proofs, not fast proofs (tho they should be fast enough). We care about L2 tx fees. The speed of ZK-Rollups has no impact on its security. For the next testnet, proposers will decide which prover to use. It might be a per-block agreement or a longer agreement (kind of a partnership). We assume proposers will choose cheap proofs.

- How long a single transaction confirmation will take in Taiko? What is the difference between transaction confirmation and block finality?

There are two ways to define finalization, and people argue which one is correct. For Taiko, once the L2 block is proposed (included in the L1 block), it can't be reverted anymore. The block is finalized from that moment, as nothing can be changed. Another finalization definition states that the block is "really" finalized only when the bridge or L1 can learn the state of the block. For ZK-Rollup, it means the block is finalized when the proof for this block is posted to L1.

- What is the endgame and vision for Taiko?

As new features are still being released on Ethereum, L2s have to adjust their designs. For example, it will happen soon with blob transactions and proto-danksharding. Taiko plans to stay a ZK-Rollup, based, decentralized, permissionless, with a multi-prover. From a long-term perspective, most execution will happen on L2s while Ethereum will stay a security and settlement layer. But we're still years away from that moment.

- Could Taiko use ZK-VM?

That's definitely an interesting approach. We consider it an option for our multi-prover approach as its performance is already good enough to be used in production.

Mikko and Alex wrapped up the community call and said bye to everyone.