

Thanks [CC](#) for review and feedback.

Intro

Rollups are scaling ethereum. But the cost of rollups is still very high for many use cases. For example at the time of writing a cost of 0.02 USD was typical on l2. This is way too expensive for many use cases where users need to have transaction fees of ~0. For example OPCraft (another example is dark forest) a version of Minecraft where every operation is carried out on chain. Even after 4844 and all of data sharding has been completed its unlikely that data costs will be low enough to allow 0 cost transactions.

In this post we propose Plasma Free that supports EVM, can run any evm contract, and has a gas fee (only prover cost ~0) that is independent of l1. The only cost is prover cost.

Plasma

Plasma started as an attempt to solve two problems.

1. Data Availability
2. Execution Validity

After recent work that solves validity (zkevm's) we revisit plasma and see what is possible if we have just one problem to solve.

Plasma assumptions

Plasma assumes:

- Users watch online to see if data is unavailable
- If so they exit from an old state
- Users watch online to make sure each transaction is correct

Can we use the same assumptions and build a zkevm based system that allows us to not put data on chain ?

Plasma Free

A block producer gets transactions from users and makes blocks. They publish each block header and a proof of validity on chain but not the data. They are supposed to share the data with all users.

If they share the data everything is fine. If they don't share the data

- User places a forced transaction in the forced transaction queue.
- If the forced transaction queue is not cleared by the block producer during the forced transaction window the latest block is reverted.
- Keep reverting until the forced tx queue gets cleared.
- Continue from that point.

The fact that users need to make a forced transaction when data is unavailable means that they need to come online once during the forced transaction window. This is the key difference between rollups. The users online assumption. With zk rollup users don't need to be online, with optimistic rollup only a single honest user needs to be online. With plasma Free all users need to come online once per forced transaction window (E.g. 1 week) in the case of an attack so that they can exit.

The system depends on users being able to execute forced transactions. These transactions should only happen very rarely. The cost of such a transaction should be the same as a typical rollup transaction. So we have free transactions until data becomes unavailable and then we have the same costs as l2s.

Forced transactions

A plasma free forced transaction is very similar to an l2 forced transaction. It is an ethereum transaction signed by the user who wants to exit. Forced transactions can be batched in the same way that l2 transactions are batched. This is how our forced transaction cost can have the same cost as l2 transactions.

Conclusion

Here we introduced plasma free. This is not an l2 and not as good as an l2s. The users online assumption is a big pain point. But for some use cases this can be acceptable. An implementation of this using an already existing zkevm should be relatively easy to build with hopefully minimal zk components that need to be built.