

Participants: [Justin Drake](#) from Ethereum Foundation, [Ben Fisch](#) from Espresso Systems, [Tomasz K. Stańczak](#) from Nethermind and Flashbots, and [Daniel Wang](#) from Taiko. Moderated by [Archie](#) (a.k.a. umede) from Taiko.

## What is sequencing and who do we need it?

Sequencing is the ordering of transactions within a chain.

There should be some explicit ordering if we want to agree on the chain state given transactions as an input. As doing first tx A and then tx B is not the same

as doing first tx B and then tx A.

There is the whole spectrum of sequencer designs. For example, how it is done today at Ethereum Layer 1: there is role of proposer who has a monopoly right over the sequencing for one slot (12 sec). The proposer rotates every single slot. One can say that it is decentralized from a macro standpoint but from a micro standpoint (one slot) it is highly centralized.

Definition remark: in L2 context, we usually say “sequencing a lot” while in L1 context we use block builders and proposers system. However, both on L1 and L2 transactions are ordered by some assigned actors and sequencing design is equally relevant both to L1 and L2.

However, one should note that there are two separate aspects of sequencing: transaction ordering and order finalization.

That is, even though sequencer orders transactions, to agree on this order of transactions (to finalize the block), the rest of the network should participate. For example, in case of Ethereum (using the PBS, Proposer-Builder Separation), the transactions are ordered by a block builder, but it is proposed by a proposer and then attested by a set of attestors. That is, it is up to the proposer and attestors if the block will be included into the chain. In case of L2s (as they are today), most of them combine transaction ordering and order finalization into one role.

## Debates on whether decentralizing sequencers is important

- Escape hatches might be considered a form of decentralizing the sequencer. They can also be considered as something in between a Based Rollup and a centralized sequencer. Because if L2 sequencer transactions are censored by the L2 centralized sequencer, users can post transactions directly to L1 and after some delay they will be forced to be included by L1. So, in case of escape hatches a part of sequencing power goes to L1. And whether or not having escape hatches is enough depends on what rollup properties we care most in each specific case. If the goal is robust censorship resistance – maybe escape hatch is good enough. If we care about rollup economics, for example, things like resistance towards monopoly pricing and price discrimination, – escape hatches might be not enough.
- Another aspect (not the same as sequencer decentralization) is sequencer permissionlessness: who can join as a sequencer?
- As a part of the endgame, decentralized sequencer is important. But from a pragmatic point of view, there are other more crucial aspects that should be prioritized. For example, having fraud proofs or validity proofs on rollups with large ongoing activity.
- The goal of an Ethereum rollup is to inherit security from L1. The “security” includes both safety and liveness. Rollups give us 100% safety of Ethereum. With escape hatches, one might assume that the 95-99% of Ethereum liveness is inherited by the rollup. But in the adversarial environment, this small delta of 1-5% can be exploited. For example, if one can censor DEX transactions or oracle activity for an hour – this is a huge position of power to create a lot of disruption and toxic MEV.
- Escape hatches are used when the rollup is totally broken. That is, users just migrate to another venue and the activity on this rollup is dead forever and it completely breaks the network effect. That doesn't work in a good way if we want credible neutrality and being a platform for users for decades and centuries.
- Based Sequencing inherits both 100% of L1 security and 100% of L1 liveness.
- To decentralize the sequencer, rollups can also use their own token. In this case, the liveness depends on the economic security of this token.

## Users perspective: should users care about sequencer decentralization?

- Most users trust the centralized sequencer for fast finality (that is, not waiting until the block tx will be included into L1 block). From ux perspective, it is very attractive. But it erodes the safety of rollup. As if users trust the sequencer for finality and if sequencer equivocates it might cause double spends. It might not be an issue today. But web2 can also be not an issue today. But it creates platform risk. The bigger the network effect of the rollup is, the less incentive it has to decentralize itself down the road (as as a platform it has more and more leverage such as userbase). If users don't care about joining a rollup that is decentralized early on, one day they might find themselves in a situation when they have no choice but to subscribe to high prices. Because with no decentralization it is a monopoly and the rollup can impose whatever pricing it wants.

- User choice is somehow a derivative of the developers' and liquidity providers choice. If the latter don't trust the chain foundation, they won't deploy their product or provide their liquidity.

## Ways to decentralized the sequencer

### Based Rollup

- The first approach is Based Sequencing: the Layer 1 proposers are the sequencers. One can argue that this is the simplest design possible. The rollup developer doesn't need to do pretty much anything to enable Shared Sequencing. The only required thing is the description of the state transition function and the ability to accept rollup blocks at every single slot.
- Based Rollup inherits 100% of Ethereum liveness: no caveats, no bugs. It also provides kind of Shared Sequencing for free whenever you have two based rollups. As the Layer 1 proposer, when he gets to propose the block, it can simultaneously propose blocks for all based rollups. It unlocks synchronous composability between the rollups that, for example, allows for shared liquidity between the rollups.
- Finally, one of the most unique features of Based Rollup is credible neutrality. As in long-term rollups are going to take most transaction volume with high probability.
- Using L1 proposer as an L2 sequencer allows to remove one layer from the supply chain that might lead to cheaper transactions. But it will be fair to note that not only Based Rollup but Shared Sequencing in general is going to reduce the transaction cost. It is true as the goal of SS is to build the most economically profitable block so they will search for all possible MEV, it will be their economically rational behaviour. And they will be able to extract all types of MEV including cross-chain MEV.
- Based rollups allow building alternatively sequenced applications on top of them (for example, building a Central Limit Order Book on top of Taiko). But the opposite is not possible.

### Shared Sequencing

- Shared Sequencing is aimed at inheriting most advantages of Based Rollup without the need to be a Based Rollup. Another goal of Shared Sequencing is to provide ux as close to centralized sequencing as possible (but without a need for centralized sequencer). That is, prioritizing high throughput and fast finality. The cost of these properties is that SS requires most nodes to be always online.

Other experimental designs (such as SUAVE, a system for managing private order flows and a source of block building for other rollups)

- It is expected by Ethereum community that all possible design experiments will appear. At the same time, as we assume that most Ethereum activity will be pushed into rollups in some med-term future, the probability that it will be mostly pushed towards Based Rollups is pretty high as users are interested in inheriting 100% of Ethereum security and 100% of Ethereum liveness.

## Debates on giving the MEV to L1 (as done by Based Rollup) vs preserving it for own purpose (as done by centralized sequencers)

- If most Ethereum activity migrates to L2, the MEV volume on L2 will be huge. In case of centralized sequencers, most MEV is expected to be extracted by the sequencers. And even though some share of MEV extracted by the sequencer might flow to L1 validator as a payment for block inclusion, most MEV value will stay at L2.
- In case of a Based Rollup, a share of MEV is given to L1 validators. The question is how much of MEV is given? There are two sources of MEV: (i) congestion fees (a.k.a. base fees in EIP-1559) and "MEV". Today on Ethereum, congestion fees constitute 80% while MEV – only 20%. Within the 20% "MEV" – half of it is toxic MEV (e.g. frontrunning) and the rest is non-toxic MEV (e.g. arbitrage). As the goal is to decrease toxic MEV to ~0% (for example, through better application design such as Uniswap X), one might expect that the share of MEV will be close to 10% and congestion fees – close to 90%. So, going with Based Sequencing, the rollup might lose up to 10% of total fees as a potential profit. But it gains credible neutrality.
- Furthermore, if MEV is expected to be handled by the L2 – how is it going to handle it? What will be the split between congestion fees and "MEV"? How the fees will be designed? How the fees are redistributed between different parties (users, block builders, L1 parties, clients (e.g. wallets) etc.)

## Hybrid model of sequencing

If we want the network behaviour to be consistent, then the probability of hybrid model is pretty low. However, if we imagine a highly volatile environment (e.g. sharply changing MEV behaviour) then hybrid models might make sense. That is, the network will switch between different sequencing mechanisms depending on the current environment state.