Proposer (sequencer) decentralization is all the talk now. But what about provers? Should rollups have decentralized provers?

Here are some of the for

and against

arguments that I've come up with or found on the internet.

For prover decentralization:

1. If you have a centralized prover (like all ZK-rollups now have) and it goes down, who picks up the work? One solution is to have some sort of a pool of backup provers who could step in. In this case, the DAO or any other governing body that owns the protocol should pass an emergency vote to open the gate for other provers. However, this would introduce some delay.
2. If there's one centralized prover, all prover rewards go to its pocket. This could be mitigated by some type of a mechanism that would redistribute the prover rewards back to tokenholders. But this might mean that proving is an unprofitable activity for the rollup team that's running the prover.
3. Even the small fish could join the prover network by using resource-pooling services. I believe this would work similarly to liquid staking: You add what computational resources you have to a large pool. The pool then uses those resources to run a prover and generate validity proofs. The rewards that the prover receives are shared between those that supplied their computational resources to the pool.

On top of that, proof computation costs and requirements are expected to go down with time, allowing for a larger diversity of provers.

ZKPool seems to be building something similar but there's not that much information on it yet. Check alsoIngonyama.

Against prover decentralization:

1. Decentralized provers are unable to censor transactions and don't increase security. The worst they can do isdelay onchain finality of L2 batches.
2. Decentralizing provers would add unnecessary complexity to the whole thing.
3. Multiple provers can be working in parallel to generate a validity proof for a single block but only one can succeed. This means that other provers who tried to generate a proof for the same block have wasted their resources. Also, there's the opportunity cost.

Also, some additional things to consider:

1. What do "decentralized" and "permissionless" mean? Are they the same thing? Can a network be decentralized but not permissionless and vice versa?

These are just thoughts from my simple mind. Galaxy brains, come and show us how it's done.

Some interesting proposals/discussions:

1. ZKP's Two-Step Submission Algorithm: An Implementation of Decentralized Provers.
2. Based rollups—superpowers from L1 sequencing - #14 by fradamt - Layer 2 - Ethereum Research.