Node Operators using the Lido Protocol have been making efforts to improve Execution Layer (EL) client diversity since minority clients stabilized and matured post-Merge (I know, NOM workstream contributors bug them a LOT about it and it's [@yorickdowne](#)'s part time job in the validators chat).

While aggregate data for Q4/24 is still being crunched, it's as good a time as any to for some public commitments by Node Operators. This would be useful to signal to not only the DAO, stakers, and the wider network that NOs are cognizant of room for improvement and taking steps towards further safeguarding the network against a possible supermajority client block validity bug that might affect a (super)majority client.

As such, I'd like to propose through this thread that Node Operators make public commitments about their plans to further decrease majority client (currently geth) utilization with regards to validators that they run as a part of the Lido protocol (and in general). It's definitely also a place for those who are leading the charge (and offering help either through the maintenance of software that helps prevent such an event or through infra migration advice) to received plaudits for doing so!

I suggest the following (which, best I can tell, is already happening but twitter gets angry if you don't cc them on everything):

"Majority client" is defined as the EL client which is most widely used in Ethereum attached to validator nodes, and having a utilization rate >= 50%*

- Node Operators using the Lido protocol target < 2/3rd majority client utilization across Lido validators (in aggregate) by end of Q1 and maintain indefinitely,

- Node Operators using the Lido protocol target <= 55% majority client utilization across Lido validators (in aggregate) by end of the year with a view to gradually decrease even more,

- Node Operators target < 2/3rd majority client utilization on a per operator basis by end of Q3, with a view to progress below 55% ASAP

*

It's possible for Node Operators to use a majority client in multi-client setups where the client is used to propose blocks but not validate them, or in setups where consensus amongst multiple clients is needed, etc, so that should be taken into account. It's not a simple case of "using geth (as an example) or not", but how the client is utilized in the NO's validation setup.