Hey,

we, from Gnosis, developed a new specification for a decentralized exchange using on chain scaling with snarks. The exchange is developed as a snark-app (snapp). The scalability is achieved by outsourcing all computations into snarks and using ethereum only as a data availability insurance and execution engine for the snarks. The approach is quite similar to Roll_up project, plasma snapp and especially onchain scaling.

Here is the specification

And here is a high-level presentation:

One remarkable feature of this specification is that it is the first exchange, which scales well and has a decentralized matching engine. With the batch auction approach, anyone has the chance to provide the best prices and thereby determine how orders are matched.

Since we need to deal with huge snarks, up to 2**28~0.26 billion constraints, calculating the snarks will be quite expensive. Our estimate for the costs is roughly 1200$ using AWS and DIZK. In order to cut costs, we are introducing a proposal-challenge game: Snarks are not provided immediately, only the new state of the snapp needs to be provided by a significantly bonded party. Then, anyone can check the proposed state of the snapp and can challenge it, in case it is not correct. Of course, the challenger also needs to bond himself. If a state transition is challenged, then the challenged person needs to deliver the snark proof its correctness.

In the longterm, he hope that we can gain more scalability by merging these snapp approaches with plasma.

We are very interested in your opinion. What do you think about the snapp architecture? Do you see any improvements for the protocol?