

tldr;

I have an interesting idea of a subscription-based DEX, which may be more fair and scaleable than regular DEXes. This exchange uses streams instead of transactions /order books.

How it works:

- you want to exchange, say 1 ETH into DAI
- you create a stream that sends 1 ETH over 24 hours into a smartcontract
- you get back a stream that sends DAI to you over 24 hours.

At any given moment in time, the exchange rate $E(t)$ is determined as the ratio of total incoming ETH stream volume to total DAI stream volume. The money you ultimately get is an integral over 24 hours of $E(t)$.

An exchange like this would have some really great properties:

- it would be fair since everyone would get the same rate
- it would be much better protected against front running, since only a small portion of money is exchanged at any given moment of time
- it would be way more scaleable than regular exchanges, since you would not need to maintain order books or split your transaction into many small transactions to average out.

Note, that if the exchange rate becomes out of sync with the exchange rate on other markets, presumably there would be lots of arbitrage agents willing to fix this.

The question becomes how to implement this efficiently in Solidity ...