

The Arbitrum sequencer receives transactions from users and publishes an ordered sequence, which serves as input to the execution stage of Arbitrum. Currently the sequencer follows a first-come, first-served (FCFS) sequencing policy.

There's a lot to like about FCFS. It's simple and easy to explain. It seems intuitively fair. It minimizes latency because the policy allows each transaction to be appended to the sequence immediately on arrival.

Other rollup protocols use FCFS as well. Optimism does, and based on descriptions of other systems they seem to do so as well.

Is there an argument for an alternative policy? In particular, does someone want to advocate for a specific, implementable transaction ordering policy, based on the pros and cons of that policy vs FCFS?