

After the transition to Proof of Stake (PoS), bundles are no longer guaranteed to be placed at the start of a block; instead, they can be placed anywhere within the block, and there's no general method to ensure they are included at the beginning.

Now, suppose a smart contract X

is deployed at block n-1

. This contract has a single run()

function and a counter that tracks how many times the run()

function has been called. The contract is deployed using a private RPC.

At block n

, multiple parties will attempt to call the run()

function on contract X

. For simplicity, let's assume these parties are submitting standard transactions, not bundles.

Now, if I deploy a smart contract Y

, which calls the run()

function on X

only if

the counter on X

is set to 0, and I submit a bundle containing this transaction that is valid only for block n

, where the transaction cannot revert, would this ensure that my transaction is placed at the beginning of the block (or at least before any other transactions that attempt to call the run()

function)?

In other words:

1. When building a block, do block builders check how a bundle behaves when included in different parts of the block, or do they discard a bundle immediately if they detect that its transaction would revert?
2. If builders do test the bundle's execution at various points in the block, and they see that including my bundle at the start ensures it succeeds, does that guarantee that my bundle will be placed at the beginning of the block?
3. Given that my bundle is the only one valid for block n

, can the other transactions calling run()

on X

(which may have higher priority fees) be deferred to the next block since they can still be executed later?