The more I think about full statelessness (stateless ethereum where even the miners hold no state), the more worried I get. Here's a first post with one of the difficulties I see, I hope to follow it up with more posts in the future:

1. It's possible to construct a transaction which consumes a lot of gas before trying to access a state which the witness did not prove. If this was allowed to happen then it would be easy to DOS miners. Ethereum usually handles this situation with reverts, the transaction is included in the block and pays gas to the miner but changes no other state. (This is called "Attributability of Missing State" by this post.

2. If I can lose money for submitting a witness which is too small then miners need to be able to prove that I submitted a witness which was too small. Ethereum usually handles this situation by asking me to sign my witness.

3. If it was costless to create a giant witness then it would be easy to DOS the network. So, transaction senders should be charged for the size of their witnesses. If this is the case, then once again transaction senders will want to sign witnesses, or else a malicious state provider / miner could DOS them by attaching a massive witness to their transaction.

4. If witnesses are signed, then state providers can't unilaterally update witnesses, they need to ask the transaction submitter to resign the transaction with the updated witness.

5. Cumulatively, all of the above means that if miners reject transactions with stale witnesses, then transaction senders will be required to stay online and resubmit transactions which have been invalidated (which happens every block, since the state root is always changing).

Here's what I mean by a stale witness. Say you have two blocks, a transaction in block n

changed account 2

, modifying a few nodes in the state tree.

Now say that I create a transaction which reads from account 1:

In order to prove the value of account 1

, given the state root for block n

, the witness includes nodes 1

, 2

, and c

. However, as of block n+1

that witness is no longer valid. The miner can check that this is a valid proof for a previous block, but it has no way of proving that no transaction in block n

changed the value of account 1

.

I can think of a few ways around the above problem:

- We simply accept that transactions are only valid until a block is mined, at which point they are invalid and must be resubmitted with the new state root.
- Miners have some way of incorporating transactions with stale witnesses into the blocks they mine, even though miners hold onto no state.
- Maybe I'm wrong that the only way to commit to a witness is to add it to the signed part of the transaction.
- All of this assumes something like the "Direct Push" model fromthis post. A more complicated architecture could push responsibility for submitting correct witnesses to other network participants.