In order to communicate our ideas and designs more clearly, we need good terminology. There is an opinion that the “Stateless” is not a good term for what we are trying to design, and I tend to agree. We might need to move away from this term in our next pivot. For now lets see if “Witness” is an appropriate term. From my point of view, it is. This is why.

The way Ethereum state transition is usually described is that we have environment E

(block hash, timestamp, gasprice, etc), block B

(containing transactions), current state S

, and we compute the next state S'

like this:

S' = D(S, B, E)

where D

is a function that can be described by a deterministic algorithm, which parses the block, takes out each transaction, runs it through the state, gathers all the changes to the state, and outputs the modified state.

The same action can be viewed in an alternative way:

HS' = ND(HS, B, E)

where we have a non-deterministic algorithm ND

, which takes merkle hash HS

of the state as input, instead of the state S

. And it outputs merkle hash of S'

, which is HS'

, instead of the full modified state.

How does this non-deterministic algorithm work? It requires a so-called oracle input, or auxiliary input, to operate. This input is provided by some abstract entity (the Oracle) that knows everything that can be known, including the full state S

. For example, imagine that the first thing that the block execution does is reading balance of an account A

. Non-deterministic algorithm does not have this information, so it needs the Oracle to inject it as a piece of auxiliary input. And not only that, the non-deterministic algorithm also needs to check that the Oracle is not cheating. Essentially, the Oracle will provide the balance of A

together with the merkle proof that leads to HS

, this will satisfy the algorithm that it has the correct information and it will proceed.

Why is this kind of algorithm called non-deterministic? Because it cannot “force” the Oracle to do anything, it is completely up to the Oracle whether the algorithm will ever succeed in computing HS

'. The Oracle can completely ignore the algorithm and never provide the input, and the algorithm will just keep “hanging”. The Oracle may also provide wrong input, in which case algorithm will most probably fail (because the input will not pass the merkle proof verification). Why “most probably”? Because if the Oracle is very very powerful, it may be able to find preimage for Keccak256 (or whatever hash function we are using in the Merkle tree) and forge merkle proofs of incorrect data. Although this may happen, it is very unlikely, and the degree to which we are sure it won’t happen is called “soundness”.

What about the term “witness”? Often the auxiliary input that the Oracle provides to a non-deterministic algorithm is called “witness”. Therefore it is appropriate to call the pieces of merkle proofs that we would like to attach to blocks or transactions “witnesses”. If we look at the “Stateless” execution as a non-deterministic algorithm, then it all makes sense

“Witness” is a more general term than “merkle proof”, because there could be other types of witnesses, for example, proofs for polynomial commitments, SNARKs, STARKs, etc.

Hope this helps someone