This call didn't really veer into particularly new developments, with the exception of the discussion of two EIPs relating to allowing a "gas payer" in transactions (discussed later). Instead, it was a good re-cap and update on the ongoing work related to the Stateless Ethereum milestones.

Martin started off with a comment encouraging those in the 1x research group to engage with the discussions happening in regular ol' 1.0 development, because the group is filled with smart people who have opinions that would be good for the all core devs discussion. Something to ruminate on.

Critical Path Re-cap

The roadmap and "critical path" is starting to take definite shape, and it's likely time to bring everything together into one place, update the tech tree, and perhaps put together a more robust source of information for newcomers to these topics to get up to speed quickly and efficiently.

Working backwards, the end goal is having witnesses fully integrated into the protocol and mandatory.

In order to get there, changes will need to be made to the gas schedule to account for witness generation. Changing the gas schedule, however, introduces problems with EVM semantics and potentially backwards-incompatible changes. These are captured largely by UNGAS (and on the same path though perhaps less dramatic is Oil / Karma).

Witnesses will need to be specified, and of course they need to be small enough to be viable. This means focusing efforts on where we thing the biggest wins will be with reducing witness sizes, which right now looks like some combination of Code Merkleization and a transition to a binary trie, which Guillaume has been working on.

On the network side, it is clear that getNodeData

needs to be overcome, and there has been some progress on flushing out what the design requirements of Merry-go-Round sync are, although it's looking more and more like SNAP does much of what we had all hoped MGR would be good for, so in an ideal world we'd be able to start syncing state with SNAP for the most part and then let MRG come in at a more mature stage.

Discussion on Meta Transactions

Generally the use case of meta transactions is one where a potential break might occur if UNGAS/Oil/Karma is enacted. This isn't for sure, and there is some nuance around whether or not the priority is in preserving the user experience of meta transactions, or whether it's actually more about batch transactions and un-guarded calls.

In the case of the former, there are two EIPs newly created that address the use case of allowing someone to specify a 'gas payer', and a second related EIP that specifies a transaction envelope. Both were written by Micah:

Ethereum Improvement Proposals

EIP 2711: Separate gas payer from msg.sender

Details on Ethereum Improvement Proposal 2711 (EIP 2711): Separate gas payer from msg.sender

Ethereum Improvement Proposals

EIP 2718: Typed Transaction Envelope

Details on Ethereum Improvement Proposal 2718 (EIP 2718): Typed Transaction Envelope

Either way, more discussion needs to be had, and we will be looking in the coming weeks to bring more in from the community to weigh in on what the priorities here are with EVM changes and mitigating potential problems with legacy contracts and use cases.

Code Merkleization

Sina, who had previously been working solo on the code merkleization research, now has some help from team 1X/Pegasys in terms of data collection, as well as analysis for breaking the code up into chunks.

One thing that is very relevant to code merkleization is the witness format (discussed next) and whether the state trie is binary or hexary.

Regardless, code merkelization might be one area where there is a clear enough path forward for witness improvement that it would be worth looking in to drafting an EIP that could potentially be discussed for a hard fork after Berlin. Sina tentatively would be the person to lead that, but we can all understand if it seems daunting and he's not being put on the spot to take this on

The transition to merkleized code is not something that has been very deeply discussed, but most imagine it as a sort of “poke” thing whereby all contracts can electively merkleize themselves… but this is far from decided.

Turbo-geth APIs

Turbo geth has switched databases, and this new one (ElementDB) has a python API that is potentially very useful for the type of data collection research that many questions in the Stateless Ethereum tech tree are going to need. One can even leave Turbo-geth running and collect data from the hot database (stability not guaranteed :).

Witness Format

The witness format is in pretty stable form, with some minor and aesthetic changes trickling in. For the moment Paul is focused on creating robust and format-agnostic tests, which will pay off in a big way as the spec is finalized.

SNAP and the binary transtition

It’s looking more and more like SNAP will actually be a reasonable means of crossing the threshold in a binary transition, the idea being that clients with a flat database layout can much more easily keep two different trie formats at once, and that should be sufficient to serve the new format to the rest of the network.

Related to this sentiment is it’s also likely that Trinity will be starting to look at switching over to a flat DB layout as well. Turbo-geth needs some more docs but likely there are a lot of lessons in there that can be applied when Trinity goes for the flatDB.

That’s all folks

Apart from some re-capping and isolated small discussions that can be seen in the transcript, this Stateless Ethereum call was pretty straightforward and not game-changing in any way. Lots of folks have been focused on the same things for the past few weeks and the fruits of their labor are just starting to make themselves known. One big takaway from this week is that we’re finally ready to start thinking about the best order and process for breaking some of these more developed pieces of work into EIPs that we can start to polish.

The next call is not scheduled, but will likely be about a month out. If you would like a recording of the call, please ask Griffin or Piper, and you can find the machine transcript here