

Using ETC as a trustless gateway to ETH2.0

Currently, the planned method to becoming a validator in ETH2.0 is to deposit 32 ETH into a smart contract on the ETH1.x chain. Given that eventually the ETH1.x chain's state will just be folded into ETH2.0 as it's own EE, I think it's reasonable to assume that new validators will just deposit their funds in this EE in order to become a validator. However, now, I suspect it will be even more difficult to acquire ETH (or whatever the unit of account will be in this context) without going through an exchange, as you can't distribute newly generated coins in PoS. I'm propose a way in which, as a community, we can use to onboard users and developers to ETH2.0 using Ethereum Classic. The TLDR summary is simply to have a copy of the deposit contract on ETC. This should be easy to do with some minor changes to EVMC.

Why Ethereum Classic and not Bitcoin Cash?

The Ethereum Classic and Ethereum communities have been doing a lot of collaboration this past year. It has led to many important projects such as EVM-LLVM. We can leverage this relationship into ETH2.0. Moreover, the Ethereum Classic blockchain has the same foundation as the current Ethereum blockchain. So, there's an element of familiarity that can be leveraged to support such an endeavor. Also, the Ethereum Classic community is more conservative in its approach to blockchain upgrades. Despite the 51% attack earlier this year, they have been working on improving the chain. Finally, mining on Ethereum Classic is very reasonable for small players. Instead of having to go through an exchange, users new to not only Ethereum but blockchains can easily start mining and earning ETC.

How is this going to work?

I don't know but I know the main problem is converting between ETC<>ETH/BETH. We will either need to use a mechanism similar to the peace bridge relay or something like Summa's cross-chain transfer auctions.

Anyway, this is just a thought I had for a while and thought I would get some feedback from the wider community.