TL;DR:

I suggest creating a store of value Execution Environment that is only tasked with ensuring balances of ETH and possibly ERC20/ERC721 tokens can be held on shards. This allows everyone to make their own choices on how to interact with other (potentially more experimental) Execution Environments and thus minimizes the damage they can do. This encourages innovation, as the barrier to trying out a new Execution Environments becomes lower.

Rationale

Execution Environments are exciting because they allow upgrading the chain continuously, and potentially allow much more experimentation on top of a stable consensus layer, without requiring hard forks.

However, opening up “layer 1.5” poses some serious questions because it is possible to have low quality execution environments compromising the security of all their users and potentially even the whole chain (if enought value is at stake). The greatest risk comes from early execution environments that “feel” good enough, but later turn out to have significant bugs or drawbacks for the system, for example:

- An Execution Environment that’s a copy of the “official/standard” Execution Environment that does, however, not charge state rent. In the beginning, state storage will be easily available from centralized and decentralized sources, but of course it leads to the same long term problems as in Eth1
- An Execution Environment that seems fine, but turns out to have a subtle bug that lets people print money discovered after 5 years. Note even Bitcoin had an critical inflation bug as recently as 2018
- And of course any other kind of critical bug that an EE might have

Invariably, when such a bug happens, we will wonder if a hard fork could be done in order to fix the problem. It will be very difficult to draw the line which EEs would be rescued, as if we want to encourage experimentation there will be a scale of more and less trustworthy EEs.

However, to improve this and actually allow more innovation, we can add a value-holding EE (VHEE), that would be used to store value (ETH, ERC20 and ERC721). It provides all the functions of these standards it replaces and potentially more [If more token standards evolve in the future, a similar EE could be created for those].

The code for this VHEE would be sufficiently simple that it could be formally verified in its entirety, so that it will not be necessary to hard fork.

Ownership

In order to be useful, the VHEE needs to have a very flexible notion of ownership. If it only provided value transfers, then it could not be the basis for a true Ethereum ecosystem. Instead, I think the best notion of ownership is something similar to the “Pay to Script Hash” (P2SH) known in Bitcoin: An address in the VHEE would be the hash of a script that validates transactions from this address. The effective owner of the funds is thus whoever can create valid inputs for this scripts, which allows for many different kinds of ownership:

- Classical single-sig and multisig wallets
- Ownership by an address in another EE
- Ownership by multiples EE addresses, plus single-sig or multisig direct ownership
- Fallback constructions: An address in an EE that has a daily spending limit, with ultimate ownership by a private key that can revoke the addresses spending privileges at any time

The latter will require that we add a 32 byte state to each address, which can be used to store a state root to any required state by the validation script.

Any ownership by an EE ultimately fully relinquishes trust to that EE. The key is that we can create constructions where we don’t have to fully trust an EE for all our funds, and can thus interact with the EE and use its advantages without the danger of losing all funds.

For the function of some EEs, e.g. optimistic rollup, full control of the funds is central. They will therefore only accept funds that are on an address that is fully controlled by that EE. Using such an EE will always be associated with a higher risk. However, since secure cross-shard transactions will be possible with single-block latency in the new design, most users will probably want to keep most of their funds in their own control.

Advantages for interaction between EEs

Without a separate VHEE, EEs can only interact (i.e. transfer value) with other EEs that they fully trust, because trusting, for example, an inflationary EE would be catastrophic to the EE accepting funds from this EE if they are not backed. However, creating the VHEE allows one EE to easily call (a contract in) another EE with funds backed by the value EE. So even if the receiving EE has no reason to trust the calling EE, it can still be assured that the funds are now safely in its control and act accordingly.

This also avoids the centralization issue that only EEs containing a large amount of value are interesting to users. Instead, the user might be able to chose the EE for every single transaction they want to execute, and different systems can use different EEs freely as they can still easily interact with most users.

Eth1 transition

The one big exception to this is of course the Eth1 EE. Since it is not aware of the VHEE, at the start, it will naturally hold all the funds that are currently on the Eth1 chain. It is also likely not possible to make any kind of automated transition for ERC20s, as the contracts can have custom functionality that cannot simply be removed.

For the foreseeable future, we will likely have to accept that the Eth1 EE itself is also a legacy value-holding EE, likely restricted to one shard. To be more flexible with this, we can add a new kind of address to this Eth1 EE that works just like the P2SH-address in the VHEE. That would allow people to hold legacy e.g. ERC20 tokens but still be able to work with it in other Eth2 execution environments.

A more experimental idea would be to have special addresses in Eth1 that represent balances held by the VHEE, that could then also be transferred within shards. However, the problem with this is that special ERC20 contracts might allow decreasing of balances, and thus the VHEE might believe it holds a larger balance than it actually does, which would be catastrophic.