The broken incentive system of Ethereum state storage (no cost for leaving your junk around forever) has been an active topic for years but AFAIK every proposal so far has turned out to be very complicated/risky/high overhead after careful examination. Here is my latest attempt, please feel free to punch holes in it

gist.github.com

https://gist.github.com/zsfelfoldi/a207d216b3fa9ae4be6abe7a5d8e68d8

storagefee.md

A pay-for-storage model for evolving tree hashed data structures

Author: Zsolt Felfoldi (zsfelfoldi@ethereum.org)

This proposal describes a relatively lightweight and easy to maintain structure that is directly applicable for the implementation of a "storage fee" scheme on the EVM state and is also flexible enough to accomodate future storage schemes. It allows renting each account and contract storage entry separately, solving the problem of funding contracts with many user entries like ERC-20 tokens. Its economic model lets users know the exact expiration time of their entries when paying for rent. It does not need removal transactions, allows lazy garbage collection but still avoids "tree rot" (scenarios where it's not instantly clear whether a transaction is executable or not). The economic model can also be considered separately from the proposed data structure. It might require some more formal study but I believe it has good stability/fairness properties.

Data structure

We differentiate between two kinds of tree nodes:

This file has been truncated. show original

(the pricing model might be interesting independently from the data structure solution, I believe it has good stability properties)