Following DRAFT: Position paper on resource pricing here are a few though on rent.

The problem I see with rent is “who should pay for it?”.

Letting anyone pay the rent for a whole contract have the issue that the cost is not attributed to the resource consumer.

For example for a token contract, someone could split its balance into a lot of accounts, taking a large amount of storage, thus increasing the rent which must be paid for the contract.

A potential solution to it would be to have storage affected to an account which should pay for it. In the case of a token contract, the data would be stored in state[token_holder][token_contract]. Only token_contract could write there, but token_holder would have to pay the gas for it. If he doesn’t his balance would become inaccessible and he would have to pay the wake-up cost.

Separating this storage by contract, allows a token holder to choose which rents he want to pay (otherwise we would have the opposite problem, contracts writing to its storage in order to increase the account rent). In term of user experience, this could be done smoothly by putting a rent deposit when the user use the contract.

If the user fails to pay the rent only this user is affected.

In token contracts, the problem looks quite simple, but for more complex contracts, allowing a user to render the storage related to him inacessible by not paying could be problematic (concrete example: in the Kleros contract if a the staking slot of a user is not accessible, the contract would not know which account is drawn and would not be able to penalize the corresponding user for failing to vote).

A solution would be to require users to prefund an “hibernation” fee which would be used to pay the execution cost of an action when storage goes into hibernation (in the case of the Kleros contract, it would be destroy all locked tokens of this account and unstake all remaining tokens).

That would introduce new challenges in smart contract development (having to specify a hibernation procedure and not always being able to rely on storage being available).